

UNIVERSITÀ DEGLI STUDI DI SALERNO
FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI
CORSO DI LAUREA IN FISICA

Reti neurali per la rivelazione di segnali
gravitazionali nell'ambito del progetto *VIRGO*

Relatori

Ch.mo Prof. R. Tagliaferri

Ch.mo Prof. L. Milano

Ch.mo Prof. F. Barone

Candidato

Fausto Acernese

Matr. 059/001413

Correlatore

Ch.ma Prof.ssa M. Marinaro

ANNO ACCADEMICO 1998/1999

A nonno Faustino

per avermi insegnato il valore dell'ascoltare

ed a zio Filippo

per avermi mostrato l'importanza dell'essere creativi

Indice

Introduzione	1
1 Le onde gravitazionali	6
1.1 Le equazioni delle onde gravitazionali	6
1.2 Generazione di onde gravitazionali	9
1.3 Tipi di segnali gravitazionali	10
1.3.1 Segnali monocromatici	11
1.3.2 Segnali transienti	13
1.4 Come rivelare le onde gravitazionali	15
1.4.1 Esperimenti di rivelazione in corso	17
1.4.2 L'interferometro VIRGO	19
2 Analisi di segnali nel rumore	32
2.1 Caratterizzazione di una serie temporale	32
2.1.1 Trasformata di Fourier	33
2.1.2 Cross-correlazione e autocorrelazione	33
2.1.3 Convoluzione	34
2.1.4 Spettro di potenza	34
2.2 Rapporto segnale-rumore	35
2.2.1 Statistica del rumore	35

2.2.2	Matched filters	37
2.3	Dati caotici in sistemi fisici	39
2.3.1	Sistemi dinamici e Caos	40
2.3.2	Ricostruzione dello spazio delle fasi	45
3	Reti Neurali	58
3.1	Fondamenti teorici - il neurone	59
3.2	Architetture principali: reti statiche	62
3.2.1	Reti MLP	63
3.2.2	Reti RBF	64
3.3	Architetture principali: reti dinamiche	67
3.3.1	Time Delay Neural Networks	68
3.3.2	Reti ricorrenti in forma canonica	69
3.3.3	Modelli NARMAX	70
3.3.4	Modello FCHLRNN	71
3.3.5	Architettura di Elman	72
3.4	Algoritmi di apprendimento	72
3.4.1	Metodi basati sul gradiente	73
3.4.2	Algoritmo di apprendimento per FCHLRNN	74
3.4.3	Metodi di ottimizzazione	77
3.5	Apprendimento e generalizzazione	79
3.5.1	Regolarizzazione	81
3.6	Preprocessing	83
3.6.1	Normalizzazione e whitening	84
3.6.2	Principal Component Analysis	85
3.7	Bayesian learning	88
3.7.1	Apprendimento Bayesiano dei pesi della rete	90
3.7.2	Distribuzione degli output della rete e barre di errore	95

3.7.3	Integrazione sugli iperparametri α e β	96
3.7.4	Implementazione pratica	98
4	Ricerca di componenti indipendenti	102
4.1	Indipendenza statistica	103
4.2	Definizione di ICA lineare	103
4.3	Identificabilità di un modello ICA	104
4.4	Funzioni di costo	106
4.4.1	Verosimiglianza ed entropia della rete	107
4.4.2	Mutua informazione e distanza di Kullback-Leibler	108
4.4.3	Tensori di cumulativi di ordine superiore	109
4.5	Algoritmi per la ICA	110
4.5.1	Algoritmi di decorrelazione non lineare	110
4.5.2	Algoritmi per la massima verosimiglianza e stimatori infomax	111
4.5.3	Algoritmi per la Pca non lineare	111
4.5.4	Algoritmo FastIca	112
4.6	Ica con ritardo temporale	113
5	Esperimenti	118
5.1	Modellizzazione con un processo NAR	118
5.1.1	Origine e preprocess dei dati sperimentali	120
5.1.2	Setup del modello NAR	123
5.1.3	Riepilogo dei risultati	124
5.2	Modellizzazione NARMA per l'esperimento <i>VIRGO</i>	133
5.2.1	Setup del modello NARMA	135
5.2.2	Esperimenti con $F_{max} = 1024 Hz$	137
5.2.3	Introduzione del rumore di <i>creep</i>	138
5.2.4	Riduzione dello spettro a $1000 Hz$	140

5.2.5	Esperimenti con $F_{max} = 2048 Hz$	142
5.2.6	Matched filter per la rivelazione del segnale	145
5.3	Blind Source Separation per i terremoti esplosivi di Stromboli	148
5.3.1	Origine dei dati	148
5.3.2	Analisi dei dati sismici	149
5.3.3	Risultati	152
Conclusioni		157
A Back Propagation		161
B Reti con connessioni dinamiche		164
C Definizioni di cumulativi di una distribuzione		166
Ringraziamenti		168

Introduzione

La teoria della relatività generale di Einstein prevede che una distribuzione di massa con momento di quadrupolo diverso da zero possa emettere onde gravitazionali. Nel 1975 i fisici Hulse e Taylor mostrarono che la variazione del periodo di rotazione della binaria PSR1913+16 poteva essere associata alla perdita di energia a seguito di emissione di onde gravitazionali. La rivelazione sperimentale, anche se indiretta, portò a due importanti conseguenze: una ulteriore verifica della teoria della relatività generale di Einstein, e la potenziale nascita di una nuova finestra di osservazione per l'universo.

L'interazione fra le onde gravitazionali e la materia è circa 40 ordini di grandezza più debole della interazione elettromagnetica: l'unica possibilità di osservazione diretta è la rivelazione di radiazione generata da sorgenti astrofisiche. Esempi tipici sono la collisione di supernovae, la cattura di materia da parte di buchi neri e di stelle di neutroni, l'emissione gravitazionale da parte di stelle di neutroni rotanti (Pulsar) o di sistemi binari collassanti (stelle binarie coalescenti).

La rivelazione sperimentale delle onde gravitazionali apre un nuovo canale informativo il quale, complementare a quello elettromagnetico e quello neutrino, è in grado di fornire contributi fondamentali a diversi campi della fisica:

- misura delle proprietà del gravitone
- indagine sulle caratteristiche dell'universo nei primi istanti di vita: si valuta

che la radiazione gravitazionale abbia iniziato a propagarsi 10^{-42} secondi dopo il Bing Bang, quella elettromagnetica solo un milioni di anni dopo

- determinazione più accurata di alcune costanti cosmologiche come la costante di Hubble, legata alla velocità di espansione dell'universo
- studio della materia ad elevata condensazione
- analisi della dinamica dei collassi gravitazionali
- misura delle masse delle stelle di neutroni
- prova diretta dell'esistenza dei buchi neri

Queste prospettive hanno spinto diverse comunità scientifiche internazionali a finanziare esperimenti volti alla rivelazione, e successivamente alla osservazione diretta di onde gravitazionali. Una prima serie di questi esperimenti furono compiuti, intorno agli anni '60, da Weber. Egli si proponeva di rivelare onde gravitazionali attraverso un risuonatore costituito da una barra di alluminio del peso di circa una tonnellata e mezzo.

L'esperimento *VIRGO*¹, nato da una collaborazione tra gruppi di ricerca italiani e francesi, si propone di costruire a Cascina (Pisa) un rivelatore sensibile alle onde gravitazionali nel range di frequenze 10-10000 Hz. Il rivelatore consiste essenzialmente in un interferometro di tipo Michelson a doppia onda i cui specchi sono sospesi. I bracci sono lunghi 3 *Km* e disposti perpendicolarmente. L'uso di cavità di Fabry-Perot accresce la lunghezza ottica fino a 120 *Km*. I componenti ottici, posti sotto vuoto spinto, sono isolati dal rumore sismico tramite un *superattenuatore* multistadio; il vuoto spinto è stato creato anche all'interno dei tubi di connessione. La posizione e l'allineamento dei componenti ottici sono mantenuti costanti grazie

¹Il nome del progetto trae origine dall'ammasso di galassie situate alla distanza di 10 *Mpc* dalla terra

ad un complesso sistema di controllo digitale. Questa breve descrizione è sufficiente a comprendere che la costruzione dell'antenna interferometrica di *VIRGO* richiede l'impiego delle più sofisticate tecnologie, in diversi campi. Ad uno sforzo tecnologico di così vaste proporzioni deve corrispondere un pari impegno nella individuazione di opportune tecniche di elaborazione dei dati che consentano l'estrazione di informazioni utili dal segnale registrato a valle dell'antenna.

Nell'esperimento *VIRGO* è prevista una frequenza di campionamento di 20 kHz e quindi un anno di presa dati equivale a circa 2.5 Terabyte di dati. La notevole quantità di informazione può essere analizzata solo con la realizzazione di elaboratori con grandissime potenze di calcolo. Inoltre la tecnica di filtraggio dei dati per la ricerca del segnale gravitazionale presuppone la conoscenza della curva di sensibilità dell'interferometro. Poichè lo spettro di rumore del rivelatore è a larga banda e poichè varie sono le frequenze caratteristiche dei segnali gravitazionali, si può effettuare un filtraggio nel dominio delle frequenze solo ad elevatissimi costi computazionali. Questo ci spinge a lavorare nel dominio del tempo e quindi ad usare tecniche specifiche di identificazione del rumore.

Tutto questo motiva l'uso delle reti neurali: l'architettura parallela di calcolo, la struttura non lineare, la possibilità di utilizzarle *on-line* senza stimare la funzione di correlazione a priori, e la possibile implementazione hardware, rendono le reti neurali adatte ad analisi di questo tipo, parallelamente od in sostituzione dei metodi tradizionali.

In questo lavoro di tesi è presentato un approccio ibrido che incorpora concetti di dinamica non lineare e di reti neurali per modellare serie temporali, identificare sistemi non lineari complessi e per separare segnali che hanno origine da diverse sorgenti. Con l'uso di tecniche moderne per l'analisi di dati caotici, identifichiamo alcune dimensioni tipiche del processo dinamico che ha generato la serie temporale. Questo ci permette di effettuare un ottimo preprocessing dei dati, tappa importante per

una corretta implementazione di una rete neurale. Dall'analisi di diverse architetture ed algoritmi di apprendimento risulta una superiorità delle reti Time Delay con apprendimento Bayesiano rispetto alle reti ricorrenti di Elman addestrate con Back-propagation. In particolare, la tecnica di Inferenza Bayesiana per la determinazione dei pesi della rete offre diversi vantaggi: analizzando gli iperparametri abbiamo un maggiore controllo della complessità del modello, l'uso di termini di regolarizzazione nella funzione di distribuzione dei pesi evita il problema dell'overfitting e l'approccio Bayesiano offre un'automatica validazione statistica del modello.

Gli obiettivi principali del lavoro di tesi sono:

1. realizzare un modello non lineare e non parametrico per predire l'evoluzione di variabili ambientali, quali la temperatura ed il rumore acustico. Dall'analisi dei residui di predizione possono essere riconosciute anomalie che compromettono la rivelazione dei segnali gravitazionali ed effettuare una operazione di trigger per selezionare *on-line* gruppi di dati per un'analisi *off-line* più raffinata ma computazionalmente dispendiosa.
2. identificare la curva di sensibilità dell'interferometro tramite un modello non lineare e non parametrico della funzione *input-output* per effettuare una operazione di *whitening* del rumore nell'intervallo di frequenza più ampio possibile. In questo modo si effettua un primo filtraggio ed è possibile l'applicazione degli algoritmi di estrazione di segnali da rumore bianco.
3. implementare un algoritmo che separi dalle registrazioni delle variabili ambientali i segnali delle diverse sorgenti di rumore affinché migliori la fase di identificazione.

La tesi è strutturata nel modo seguente: nel capitolo I vengono presentati alcuni concetti di base sulle onde gravitazionali, sulle sorgenti, sulle tecniche di rivelazione

e sul progetto *VIRGO*. Nel capitolo II vengono illustrate le tecniche principali dell'analisi dei dati per sistemi deterministici e sistemi stocastici con alcuni cenni all'analisi di sistemi dinamici caotici. Nel capitolo III viene illustrata la base teorica delle reti neurali, delle principali architetture e degli algoritmi di apprendimento, con particolare attenzione alle tecniche di inferenza Bayesiana. Nel capitolo IV viene illustrato il problema della separazione di sorgenti non conosciute (Blind Sources Separation), con l'analisi delle componenti indipendenti (Independent Component Analysis). Infine nel capitolo V sono presentati gli esperimenti effettuati per validare le tecniche di analisi proposte nella tesi.

Capitolo 1

Le onde gravitazionali

1.1 Le equazioni delle onde gravitazionali

Uno dei concetti fondamentali della teoria della relatività speciale è la distanza infinitesima ds tra due punti nello spazio degli eventi, definita dall'espressione:

$$ds^2 = dx^2 + dy^2 + dz^2 - c^2 dt^2 \quad (1.1)$$

riformulata, introducendo il quadrivettore spazio-tempo $dx^\mu = (d\mathbf{x}, c^2 dt^2)$, come:

$$ds^2 = \eta_{\mu\nu} dx^\mu dx^\nu \quad (1.2)$$

dove $\eta_{\mu\nu}$ è la matrice metrica di Minkowski in coordinate cartesiane

$$\eta_{\mu\nu} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (1.3)$$

Lo stesso ente fisico è trasposto nella teoria della relatività generale, con la differenza che lo spazio-tempo non è necessariamente piatto, come descritto dalla metrica di Minkowski, ma presenta una curvatura, manifestazione del campo che noi chiamiamo

gravitazionale. L'espressione più generale dell'intervallo spazio-temporale diventa allora

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu \quad (1.4)$$

dove tutte le informazioni fisiche sono contenute nel tensore metrico $g_{\mu\nu}$.

Posto $G = 1$ e $c = 1$, le equazioni di campo che legano la metrica dello spazio-tempo, con il tensore energia-impulso sono [1]:

$$G_{\mu\nu} = 8\pi T_{\mu\nu} = 8\pi(\mathbf{P}c, E) \quad (1.5)$$

dove l'informazione sulla metrica è contenuta nel tensore di Einstein

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R \quad (1.6)$$

essendo $R_{\mu\nu}$ e R rispettivamente il tensore e lo scalare di Ricci e $g_{\mu\nu}$ il suddetto tensore metrico.

Nel caso di deboli perturbazioni dello spazio-tempo piatto è possibile sviluppare linearmente il tensore $g_{\mu\nu}$ intorno al tensore metrico di Minkowski:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \quad \text{con } \|h_{\mu\nu}\| \ll 1 \quad (1.7)$$

Possiamo così riscrivere l'equazione 1.5 in termini del tensore $h_{\mu\nu}$.

Nell'ipotesi di perturbazioni deboli, le equazioni non lineari di Einstein si possono linearizzare e si può effettuare una speciale scelta di *gauge*, detta *transverse traceless gauge* o *TT-gauge*, in modo da rendere $h_{\mu\nu}$ un tensore trasverso ¹ a traccia nulla che obbedisce alla equazione delle onde:

$$\partial^\mu \partial^\nu h_{\mu\nu} = \left(\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) h_{\mu\nu} = 0 \quad (1.8)$$

Così gli elementi di $h_{\mu\nu}$ hanno la forma $h(i(\omega t - \mathbf{k} \cdot \mathbf{x}))$, con $\omega = c|\mathbf{k}|$, e rappresentano un'onda piana che si propaga nella direzione $\hat{\mathbf{k}}$ con velocità c .

¹Un tensore $h_{\mu\nu}$ è detto trasverso se $\partial^\mu h_{\mu\nu} = 0$

Consideriamo ora il caso di un'onda che si propaga nella direzione dell'asse \hat{z} . La scelta della TT-gauge implica che $h_{\mu\nu}$ deve essere della forma:

$$h_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a & b & 0 \\ 0 & b & -a & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.9)$$

e quindi possiamo scrivere l'onda come somma di due componenti: $h = a\hat{h}_+ + b\hat{h}_\times$, con

$$\hat{h}_+ = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \hat{h}_\times = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.10)$$

I tensori di base h_+ e h_\times rappresentano le due direzioni ortogonali di polarizzazione.

L'idea che sta alla base della rivelazione di onde gravitazionali è quella di misurare la variazione di distanza tra particelle investite da una onda gravitazionale [2]. Per vedere l'effetto sulla distanza tra masse test a seconda del tipo di polarizzazione è utile la rappresentazione di un insieme di masse disposte su di una circonferenza [3], come si vede in figura 1.1.

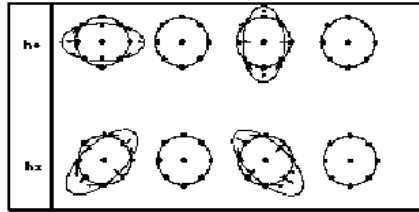


Figura 1.1: Effetto di un'onda gravitazionale su un insieme di masse test

1.2 Generazione di onde gravitazionali

Per comprendere i meccanismi di generazione di onde gravitazionali possiamo estendere le conoscenze che abbiamo acquisito dall'elettrodinamica classica secondo le quali l'emissione di onde elettromagnetiche può essere espressa in termini di un potenziale ritardato. Ma, come per l'elettrodinamica classica, è spesso più pratico lavorare nell'approssimazione di espansione in multipolo, valida nel limite di dimensioni della sorgente piccole rispetto alla lunghezza d'onda della radiazione emessa.

Ricordiamo che il contributo dominante al campo di radiazione elettromagnetica proviene dalla variazione temporale del momento di dipolo elettrico:

$$\mathbf{E} = \frac{1}{Rc^2} \left(\ddot{\mathbf{d}} \times \hat{\mathbf{n}} \right) \times \hat{\mathbf{n}} \quad (1.11)$$

dove R è la distanza tra la sorgente ed il punto di osservazione, con direzione data dal versore \mathbf{n} , e \mathbf{d} è il momento di dipolo, definito da

$$\mathbf{d} \equiv \int dV \mathbf{r} \rho_q(\mathbf{r}) \quad (1.12)$$

dove ρ_q è la densità di carica e l'integrale è esteso a tutto il volume della sorgente.

La grande somiglianza fra la legge di gravitazione di Newton e la legge di Coulomb ci induce a cercare varie analogie anche nella loro estensione al caso relativistico. Così possiamo definire un momento di dipolo gravitazionale:

$$\mathbf{d}_g \equiv \int dV \mathbf{r} \rho(\mathbf{r}) \quad (1.13)$$

dove $\rho(\mathbf{r})$ rappresenta la densità di massa. Ma la legge di conservazione del momento richiede che \mathbf{d}_g rimanga costante e quindi, essendo nulle tutte le derivate, non abbiamo emissione di radiazione.

Definiamo un analogo gravitazionale del momento magnetico di dipolo:

$$\mu_g = \int dV \mathbf{r} \rho(\mathbf{r}) \times \mathbf{v} \quad (1.14)$$

ma poiché μ_g è costante in base al teorema di conservazione del momento angolare, anche in questo caso non abbiamo radiazione emessa.

Dobbiamo allora analizzare distribuzioni di massa con momento di quadrupolo diverso da zero. Definiamo, indipendentemente dall'approssimazione di perturbazione debole, il momento di quadrupolo ridotto:

$$I_{\mu\nu} \equiv \int dV (x_\mu x_\nu - \frac{1}{3} \delta_{\mu\nu} \mathbf{r}^2) \rho(\mathbf{r}) \quad (1.15)$$

Possiamo ora scrivere l'analogo gravitazionale dell'equazione 1.11

$$h_{\mu\nu} = \frac{2G}{Rc^4} \ddot{I}_{\mu\nu} \quad (1.16)$$

dove il secondo membro dell'equazione è valutato al tempo ritardato $t - R/c$. Notiamo infine che le onde gravitazionali trasportano energia la cui la potenza è:

$$\frac{dE}{dt} \propto \left\langle \frac{d^3 I_{\mu\nu}}{dt^3} \frac{d^3 I^{\mu\nu}}{dt^3} \right\rangle \quad (1.17)$$

1.3 Tipi di segnali gravitazionali

I segnali gravitazionali si possono suddividere secondo il loro comportamento temporale in

- **periodici** se sono composti da segnali sinusoidali con frequenza lentamente variabile rispetto ai tempi di osservazione;
- **transienti** se durano poco rispetto al tempo tipico di osservazione;
- **stocastici** se hanno fluttuazioni stocastiche e durano per tempi lunghi rispetto ai tempi di osservazione.

Per ciascun tipo di segnale esiste una probabile sorgente ad esso associabile. I fenomeni di coalescenza di stelle binarie o di supernovae generano segnali transienti,

le pulsar segnali periodici. Segnali stocastici sono in genere associati ad un fondo di segnali scorrelati come quelli emessi da un insieme di pulsar, oppure dal background primordiale di onde gravitazionali, analogo al fondo cosmico di microonde.

1.3.1 Segnali monocromatici

Sorgenti astrofisiche

L'esplosione di una supernova produce un oggetto stellare che, se possiede una massa superiore al limite di Chandrasekhar (1.4 masse solari M_{\odot}^2), diventa una stella di neutroni. Se la massa è più piccola di tale limite da luogo ad una nana bianca; se la massa è più grande di $2 M_{\odot}$ si genera un meccanismo di collasso che porta ad un buco nero.

Per una stella di neutroni di massa vicina al limite di Chandrasekhar, le dimensioni tipiche sono un diametro di $20 - 30 \text{ km}$ e densità centrale di $10^{14}-10^{15} \text{ g/cm}^3$. La maggior parte delle stelle di neutroni sono osservabili grazie alla emissione elettromagnetica ad esse associata che le identifica come *pulsar*. Altra caratteristica delle pulsar è il forte campo magnetico, da 10^{12} gauss per le pulsar giovani fino a 10^8 gauss per le pulsar più vecchie. Al di fuori della stella il campo magnetico domina tutti i processi fisici ed il dipolo non allineato con l'asse di rotazione genera un'onda elettromagnetica nella banda delle frequenza radio, responsabile della principale perdita di energia di rotazione, e quindi dei rallentamenti osservati.

Pulsar note

La maggior parte delle pulsar note hanno periodi di rotazione che vanno da 0.25 a 4 secondi, ed esso aumenta col tempo a causa della perdita di energia. La rotazione rallenta su tempi dell'ordine di $10^6 - 10^8$ anni. La ricerca delle pulsar su tutta la

$^2M_{\odot} = 1.98892 \cdot 10^{30} \text{ Kg}$

sfera celeste è limitata dalla sensibilità degli strumenti. Dalla statistica del campione di pulsar disponibile sono state dedotte alcune caratteristiche della popolazione. La popolazione di pulsar nella nostra galassia è di circa $10^5 - 10^6$ pulsar, concentrate nel piano galattico entro uno spessore di 1 Kpc ³ ed entro una distanza radiale di circa 10 Kpc dal centro. Essa hanno velocità molto alte e si allontanano dal piano galattico con una velocità di 100 km/s .

Descrizione del segnale gravitazionale

Ricordiamo che affinché una pulsar emetta onde gravitazionali, deve possedere un'asimmetria nell'asse di rotazione. I due modi h_+ e h_\times dell'onda gravitazionale emessa da una pulsar, con asse di rotazione deviato di un angolo α rispetto all'asse di simmetria della sua distribuzione di massa, sono:

$$h_+ = h_0 \sin \alpha \left[\frac{1}{2} \cos \alpha \sin \phi \cos \phi \cos \Omega_p t - \sin \alpha \frac{1 + \cos^2 \phi}{2} \cos 2\Omega_p t \right] \quad (1.18)$$

$$h_\times = h_0 \sin \alpha \left[\frac{1}{2} \cos \alpha \sin \phi \sin \Omega_p t - \sin \alpha \cos \phi \sin 2\Omega_p t \right] \quad (1.19)$$

dove ϕ è l'angolo di inclinazione della linea di vista rispetto all'angolo di rotazione e Ω_p è la frequenza di rotazione della pulsar.

Si caratterizza l'asimmetria nella distribuzione di massa con un parametro ϵ , definito come

$$\epsilon = -\frac{3}{2} \frac{I_{zz}}{I} \quad (1.20)$$

con I momento di inerzia rispetto all'asse di rotazione e I_{zz} momento di inerzia rispetto all'asse di simmetria. Allora una stima dell'ampiezza dell'onda gravitazionale in funzione del momento di inerzia, dell'ellitticità ϵ e del periodo rivoluzione P è data da

$$h_0 = \frac{16\pi^2 G}{c^4} \frac{I\epsilon}{P^2 r} \approx 10^{-22} \div 10^{-20} \quad (1.21)$$

³ $1\text{ Kpc} = 3.08 \cdot 10^{25}\text{ m}$

essendo r la distanza tra la stella ed il rivelatore. Si può notare allora come l'emissione di radiazione da pulsar è tanto più grande quanto più questa è vicina, quanto più risulta asimmetrica nella distribuzione di massa rispetto all'asse di rotazione e quanto più ruota velocemente.

1.3.2 Segnali transienti

Sorgenti astrofisiche

Tra tutte le sorgenti di onde gravitazionali, la coalescenza di stelle binarie compatte è quella più studiata e compresa, almeno nell'approssimazione di masse puntiformi.

Due corpi massivi rotanti intorno al comune centro di massa emettono onde gravitazionali. A causa di questa perdita di energia, avviene un avvicinamento a spirale delle due stelle fino alla fusione finale. Le binarie più interessanti possono essere formate da due stelle di neutroni (NS-NS), da due buchi neri (BH-BH), o da una stella di neutroni ed un buco nero (NS-BH). Un esempio di un sistema NS-NS è la binaria PSR1913+16. Il periodo di rivoluzione di tale sistema è di 8 ore. A causa della emissione di onde gravitazionali le due stelle spiraleggiano e si prevede che la coalescenza avverrà tra 10^8 anni.

Descrizione del segnale gravitazionale

Considerando gli istanti iniziali del fenomeno di coalescenza, si possono trascurare gli effetti relativistici, e si può utilizzare il formalismo di quadrupolo.

Indicando con $M = M_1 + M_2$ la massa totale e con $\mu = M_1 M_2 / M$ la massa ridotta, si può determinare l'energia dissipata come

$$\frac{dE}{dt} = -\frac{32G}{5c^5} \mu^2 r^4 \omega^6 = -\frac{32G^{7/3}}{5c^5} (\mu M^{2/3})^2 \omega^{10/3} \quad (1.22)$$

Ricordando che l'energia totale di un sistema binario è

$$E_{tot} = -\frac{GM\mu}{2r} \quad (1.23)$$

si ottiene, nell'approssimazione di masse puntiformi, l'equazione

$$\frac{1}{3}\mu(GM)^{2/3}\omega^{-1/3}\dot{\omega} = \frac{32G}{5c^2}(GM)^{4/3}\omega^{10/3} \quad (1.24)$$

e quindi una relazione per la variazione delle frequenze:

$$\frac{\dot{\omega}}{\omega} = \frac{96G^{5/3}}{5c^5}\mu M^{2/3}(\omega)^{8/3} \quad (1.25)$$

Integrando questa equazione, dopo aver sostituito $\omega = 2\pi\nu$, possiamo scrivere

$$\nu(t) = 100 \left(\left(\frac{\nu_0}{100} \right)^{-8/3} - \alpha K t \right)^{-3/8} \text{ Hz} \quad (1.26)$$

dove ν_0 è la frequenza al tempo t e $\alpha \approx 0.333$ una costante. L'ampiezza dell'onda gravitazionale risulta allora

$$h = \frac{4G^{5/3}}{rc^4}\mu M^{2/3}\omega^{2/3} \quad (1.27)$$

Le due componenti del segnale gravitazionale sono

$$h_+ = h \cos \varphi(t) \quad (1.28)$$

$$h_\times = h \sin \varphi(t) \quad (1.29)$$

con fase pari a:

$$\varphi(t) = \pi \int \nu(t) dt \quad (1.30)$$

Con queste approssimazioni il segnale di una coalescenza binaria è dato da

$$h_+ = h \cos(\varphi_0 - 2\pi\nu_0\chi(t)) \quad (1.31)$$

$$h_\times = h \sin(\varphi_0 - 2\pi\nu_0\chi(t)) \quad (1.32)$$

dove per h e χ valgono le relazioni:

$$h = \alpha \frac{\nu^{2/3}}{r} \quad (1.33)$$

$$\chi(t) = \frac{8}{5}\tau \left(1 - \frac{t}{\tau} \right)^{8/3} \quad (1.34)$$

con τ uguale al tempo in secondi fra il momento in cui la frequenza è ν_0 e la fine della coalescenza.

Durante l'ultima fase della coalescenza l'approssimazione di debole perturbazione non vale più. In tal caso si usa un altro formalismo, detto post-newtoniano, che porta a determinare correzioni per tener conto degli effetti relativistici sempre più non lineari. Tale problema non verrà trattato in questo lavoro.

1.4 Come rivelare le onde gravitazionali

Come visto nel capitolo precedente, per rivelare un'onda gravitazionale dobbiamo valutare la variazione di distanza fra due masse test poste molto lontane fra loro. È ragionevole effettuare tale misura considerando il tempo impiegato da un raggio luminoso per percorrere tale distanza. Un "Toy Model" di rivelatore può essere un interferometro del tipo Michelson con la differenza che gli specchi non sono connessi ad una singola struttura rigida.

Consideriamo allora la versione più semplice di un possibile esperimento: un separatore di fascio (beam splitter) costituente una massa libera al vertice di una struttura ad L e due specchi, anch'essi masse libere, alla fine dei bracci della L. Per vedere come può lavorare tale strumento consideriamo che, secondo la teoria speciale della relatività, un raggio luminoso ha velocità c in ogni sistema di riferimento inerziale. Matematicamente questo significa che un raggio luminoso connette tutti i punti separati da un intervallo spazio-temporale nullo ($ds = 0$).

Immaginiamo, senza perdere di generalità, che i due bracci dell'interferometro siano allineati lungo gli assi x e y , con il beam splitter all'origine. Allora, poiché il percorso della luce in ciascun braccio avrà o la componente x o la componente y non nulli, dobbiamo considerare solo le componenti 11 e 22 della matrice metrica.

Analizziamo prima il braccio lungo l'asse x . Si avrà che:

$$\begin{aligned} ds^2 &= g_{\mu\nu} dx^\mu dx^\nu \\ &= (\eta_{\mu\nu} + h_{\mu\nu}) dx^\mu dx^\nu \\ &= -c^2 dt^2 + (1 + h_{11}(i(\omega t - \mathbf{k} \cdot \mathbf{x}))) dx^2 \end{aligned} \quad (1.35)$$

Allora l'effetto di un'onda gravitazionale è modulare la distanza fra due punti posti ad una distanza fissa dx con una funzione h_{11} .

Possiamo valutare il tempo impiegato dalla luce a percorrere la distanza fra il beam splitter e lo specchio alla fine del braccio \hat{x} :

$$\int_0^{\tau_{out}} dt = \frac{1}{c} \int_0^L \sqrt{1 + h_{11}} dx \approx \frac{1}{c} \int_0^L \left(1 + \frac{1}{2} h_{11}\right) dx \quad (1.36)$$

dove abbiamo effettuato uno sviluppo al primo ordine per $h_{11} \ll 1$. Scriviamo una equazione simile per il tratto dallo specchio al beam splitter:

$$\int_{\tau_{out}}^{\tau_f} dt = -\frac{1}{c} \int_L^0 \sqrt{1 + h_{11}} dx \approx -\frac{1}{c} \int_L^0 \left(1 + \frac{1}{2} h_{11}\right) dx \quad (1.37)$$

ed otteniamo il tempo totale di percorrenza nel braccio:

$$\tau = \frac{2L}{c} + \frac{1}{2c} \int_0^L h_{11}(i(2\pi ft - \mathbf{k} \cdot \mathbf{x})) dx - \frac{1}{2c} \int_L^0 h_{11}(i(2\pi ft - \mathbf{k} \cdot \mathbf{x})) dx \quad (1.38)$$

Gli integrali possono essere valutati esprimendo l'argomento come funzione della posizione del fronte d'onda che interagisce con il beam splitter al tempo $t = 0$ ed attraversa tutto l'apparato. Così dobbiamo sostituire $t = x/c$ per il primo integrale e $t = (2L - x)/c$ per il secondo.

Una espressione simile può essere ricavata anche per il raggio che attraversa il braccio lungo l'asse \hat{y} , con la differenza che ora bisogna considerare la componente h_{22} anziché h_{11} e sostituire $t = y/c$ per il primo integrale e $t = (2L - y)/c$ per il secondo. L'output dell'interferometro indica il relativo shift di fase dei due fasci che generano una figura di interferenza quando si ricongiungono sul beam splitter.

Lo shift di fase è semplicemente $\frac{2\pi c}{\lambda}$ volte la differenza del tempo di percorrenza del raggio luminoso nei due bracci. Possiamo allora scrivere una espressione generale che lega la differenza temporale di percorrenza nei bracci con la direzione di arrivo, la frequenza e la polarizzazione dell'onda gravitazionale. In figura 1.2 si vede l'effetto di un'onda gravitazionale polarizzato h_+ sulle masse libere dell'interferometro.

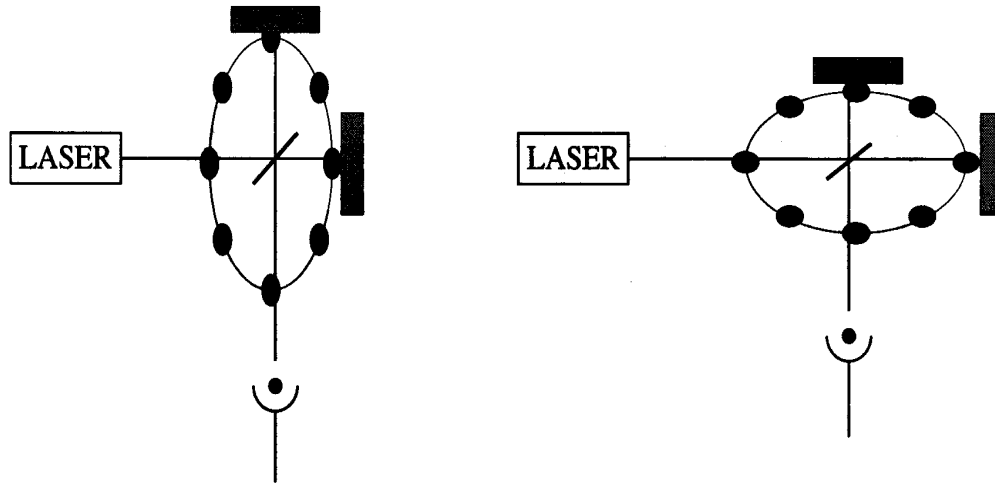


Figura 1.2: Effetto di un'onda gravitazionale sulle masse libere dell'interferometro

1.4.1 Esperimenti di rivelazione in corso

Essenzialmente i tipi di rivelatori sono di due tipi:

rivelatori risonanti: sono stati i primi rivelatori costruiti. Si basano sull'idea che al passaggio di un'onda gravitazionale una barra di metallo, con una propria caratteristica frequenza di risonanza, comincia ad oscillare variando la lunghezza L di una quantità δL . Se la frequenza dell'onda è nella banda passante della barra risonante, è possibile misurare la variazione di lunghezza $\delta L \simeq hL/2$, con h pari all'ampiezza dell'onda gravitazionale.

Il precursore di questo tipo di esperimenti fu Weber che nel 1966 costruì una barra di alluminio di 1.5 metri e di massa di 1.4 tonnellate con frequenza di risonanza di 1660Hz. Da allora sono stati fatti molti progressi per migliorare il rapporto segnale rumore, aumentando il fattore di qualità $Q = \omega/\Delta\omega$, raffreddando la barra in elio liquido per ridurre il rumore termico, sospendendo la barra per ridurre il rumore sismico. Difetto principale dei rivelatori risonanti è sicuramente la quasi monocromaticità della banda di frequenza intorno a quella di risonanza che può essere amplificata. Attualmente i progetti in corso sono:

- AURIGA, NAUTILUS, EXPLORER, ALTAIR in Italia
- ALLEGRO negli USA
- NIOBE in Australia

rivelatori interferometrici: consistono essenzialmente in un interferometro di Michelson con caratteristiche simili a quelle descritte nel paragrafo 1.4. Lo schema è il seguente: un fascio di luce, inviato su uno specchio semitrasparente separatore di fasci, è diviso in due. I fasci così ottenuti si propagano perpendicolarmente e sono riflessi dagli specchi posti all'estremità dell'interferometro. La loro ricombinazione avviene sullo specchio semiriflettente (beam splitter). La differenza di fase $\delta\phi$ tra i due fasci è indicativa della differenza di lunghezza dei bracci. Se lo strumento viene tarato opportunamente, l'interferenza è totalmente distruttiva e l'uscita è sulla frangia scura. Tutte le variazioni di lunghezza di uno dei bracci rispetto all'altro causano la presenza di luce all'uscita dell'interferometro.

Un rivelatore interferometrico offre i vantaggi di una maggiore sensibilità e soprattutto un'accessibilità ad una banda di frequenza più ampia. Questo ha portato alla progettazione di molti esperimenti, tra cui ricordiamo:

- LIGO: due rivelatori con bracci di 4000 metri, uno alla Hanford Reservation e l'altro alla Livingston Parish, negli USA.

- GEO: interferometro con bracci di 600 metri ad Hannover, in Germania.
- TAMA: interferometro con bracci di 300 metri a Tokyo, in Giappone.
- VIRGO: interferometro con bracci di 3000 metri, in Italia.

1.4.2 L'interferometro VIRGO

VIRGO è una collaborazione fra gruppi di ricerca di Italia e Francia per la realizzazione a Cascina, in provincia di Pisa, di un interferometro per la rivelazione di onde gravitazionali (vedi figura 1.3).

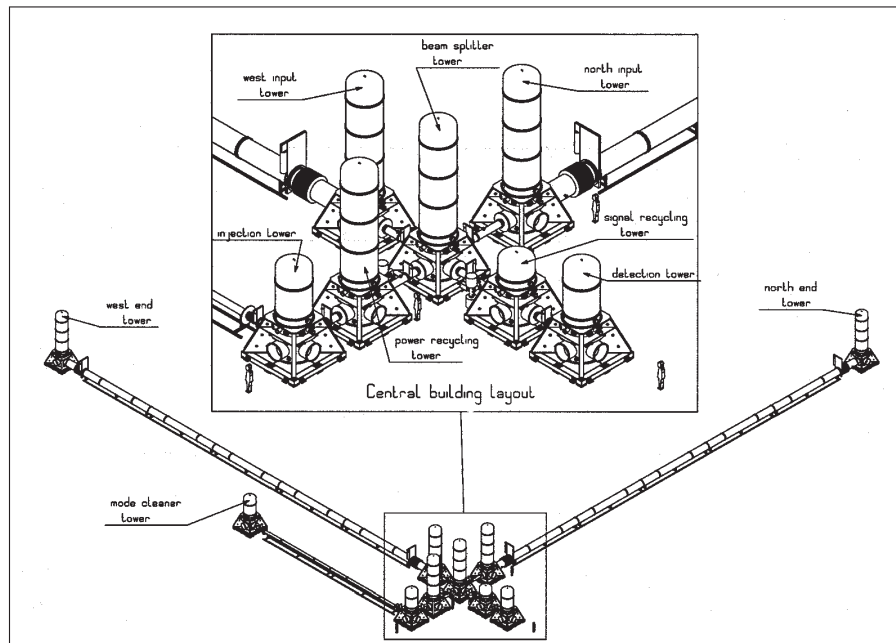


Figura 1.3: Modello dell'interferometro VIRGO

L'interferometro VIRGO è un interferometro a doppia onda del tipo Michelson avente due bracci perpendicolari della stessa lunghezza fisica di $L = 3 \text{ Km}$. L'uso di una cavità di Fabry-Perot in ogni braccio genera un cammino equivalente di 120 km. La luce è generata da un laser Nd:YAG a frequenza singola ($\lambda = 1.06 \cdot 10^{-6} \text{ m}$) con

una potenza di circa 20 Watts, che diventano circa 1 kW utilizzando un'adeguata tecnica di recupero dell'energia persa dall'uscita non utilizzata dall'interferometro (vedi figura 1.4). Le componenti ottiche dell'interferometro sono situate in vuoto spinto ed i fasci laser viaggiano in tubi sotto vuoto del diametro di 1 m.



Figura 1.4: Schema ottico di VIRGO

Ogni componente ottica è isolata dal rumore sismico da un attenuatore multistadio e multidimensionale (vedi figura 1.5).

L'attenuatore consiste in una catena di cinque stadi più una struttura iniziale. Il principio di funzionamento si basa sul fatto che quando si è lontani superiormente dalla frequenza di risonanza di un pendolo, la funzione di trasferimento è proporzionale a $(\nu_0/\nu)^2$, dove ν_0 è la frequenza di risonanza e ν è la frequenza della forzante. Per una catena di N pendoli si ha una attenuazione del rumore sismico, agente sul punto di sospensione rispetto alla base, proporzionale a $(\nu_0/\nu)^{2N}$. Poiché le componenti verticali e quelle orizzontali del moto sismico nel sito di Cascina sono

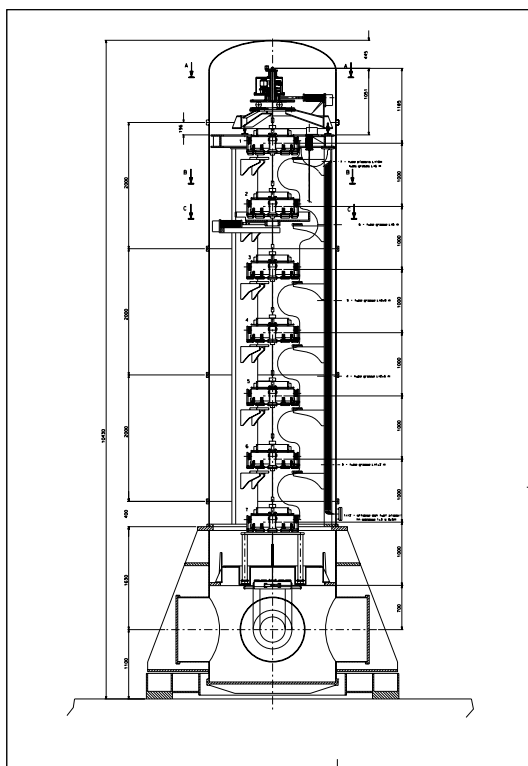


Figura 1.5: Filtro multistadio per attenuare il rumore sismico

confrontabili, per evitare un effetto di accoppiamento, è necessario anche un sistema di attenuazione per il moto verticale. A tale scopo ciascun filtro della catena è costituito da un sistema di lame metalliche e magneti il cui effetto combinato permette di raggiungere il grado di sensibilità desiderato. L'accoppiamento tra gradi di libertà orizzontali e rotazionali viene invece ridotto con un momento di inerzia di ciascun filo molto grande e riducendo i bracci del momento torcente. Per raggiungere una buona sensibilità a 10 Hz il superattenuatore è stato progettato come una catena di 5 pendoli, ciascuno lungo un metro. In figura 1.6 si può vedere la funzione di trasferimento.

Attualmente l'esperimento VIRGO è progettato per la rivelazione a larga banda, in modo da indagare su differenti tipi di sorgenti conosciute, come pulsar e binarie

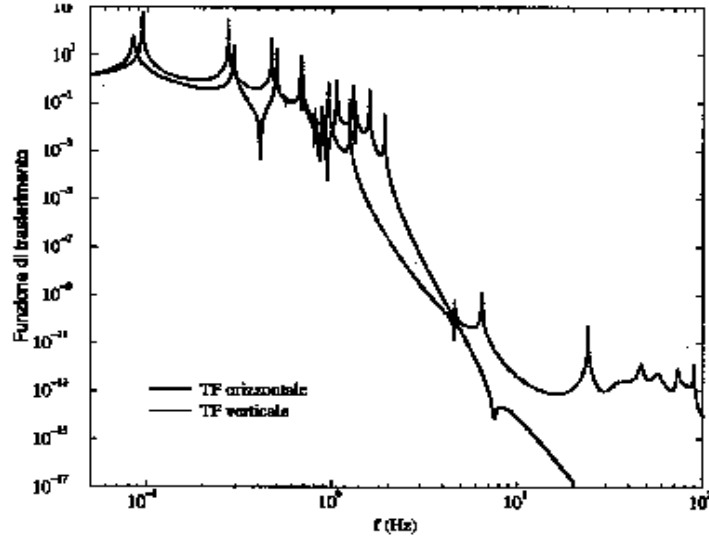


Figura 1.6: Funzione di trasferimento del filtro sismico

coalescenti, nella nostra galassia od in quelle vicine. Per ottenere una maggiore frequenza di eventi è necessario potenziare lo strumento per raggiungere l'ammasso stellare VIRGO, che contiene circa 300 grandi galassie. Questo significa un aumento di sensibilità in h pari a $10^{-23}/\sqrt{Hz}$ al di sopra dei 100 Hz e una sensibilità di $10^{-21}/\sqrt{Hz}$ intorno ai 10 Hz. Ciò permetterebbe l'analisi di pulsar note aventi potenzialmente frequenza di emissione nel gravitazionale poco maggiore di 10 Hz.

Spettro di rumore VIRGO

Nelle sezioni precedenti abbiamo visto come il segnale gravitazionale sia debole anche in sorgenti relativamente vicine. Questo rende di primaria importanza la conoscenza approfondita delle sorgenti di rumore che contribuiscono alla curva di sensibilità dello strumento nella banda di frequenze in cui vogliamo cercare il segnale.

Nello stimare le sorgenti di rumore supponiamo che il processo stocastico associato a ciascuna di esse sia gaussiano e che la statistica sia invariante per traslazioni

temporali. Definiamo allora lo spettro nel continuo come

$$S_h(\nu) = \lim_{T \rightarrow \infty} \left\langle \left| \frac{1}{\sqrt{T}} \int_{-T/2}^{T/2} h(t) e^{2\pi i \nu t} dt \right|^2 \right\rangle \quad (1.39)$$

essendo $h(t)$ la serie temporale di rumore dell'apparato, e la sensibilità come

$$h(\nu) \equiv \sqrt{2S_h(\nu)} \quad (1.40)$$

dove il fattore due è necessario se prendiamo solo lo spettro a frequenza positive.

Rumore sismico: il rumore sismico domina la sensibilità dell'apparato a basse frequenze, nell'intervallo fino a 10 Hz. Il rumore sismico misurato nel sito di Cascina può essere parametrizzato dal modello

$$|x(\nu)| = \begin{cases} \frac{10^{-10}}{\nu^3} & \text{per } \nu < 0.05 \text{ Hz} \\ 10^{-6} & \text{per } 0.05 < \nu < 0.3 \text{ Hz} \\ \frac{10^{-7}}{\nu^2} & \text{per } \nu > 0.3 \text{ Hz} \end{cases} \quad (1.41)$$

Shot Noise: la sensibilità con la quale possiamo rivelare l'onda gravitazionale dipende dalla variazione minima di potenza ottica all'uscita dell'interferometro da cui risaliamo ad una differenza di cammino ottico percorso dal raggio laser. Misurare la potenza ottica equivale a contare il numero di fotoni che giungono al rivelatore in un dato intervallo di tempo.

La statistica che regola la distribuzione dei fotoni è una statistica di Poisson, che nel limite dei grandi numeri è approssimata da una Gaussiana con deviazione standard $\sigma = \sqrt{\langle N \rangle}$. Se assumiamo come punto di lavoro quello in cui la potenza in uscita è la metà della potenza in entrata all'interferometro, allora il numero medio di fotoni in uscita sarà

$$\langle N \rangle = \frac{\lambda}{2\pi\hbar c} \frac{P_{in}}{2} \tau \quad (1.42)$$

dove λ è la lunghezza d'onda del laser e τ è l'intervallo di tempo in cui si svolge la misura. La fluttuazione nella differenza delle posizioni delle masse test è

$$\sigma_{\delta L} = \frac{\sigma}{\langle N \rangle} \left(\frac{1}{P_{out}} \frac{dP_{out}}{dL} \right)^{-1} = \sqrt{\frac{\hbar c \lambda}{4\pi P_{in} \tau}} \quad (1.43)$$

La fluttuazione relativa di sensibilità è data da

$$\sigma_h = \frac{1}{L_e} \sqrt{\frac{\hbar c \lambda}{4\pi P_{in} \tau}} \quad (1.44)$$

Non c'è nessuna scala caratteristica di frequenze per questo rumore che può considerarsi con ampiezza di densità spettrale costante e pari a

$$h(\nu) = \frac{1}{L_e} \sqrt{\frac{\hbar c \lambda}{4\pi P_{in}}} \quad (1.45)$$

Da questa equazione è evidente che per migliorare la sensibilità conviene avere una lunghezza ottica più grande possibile. Questo si ottiene nell'esperimento con le cavità di Fabry-Periot, che rendono la lunghezza equivalenti dei bracci pari a

$$L_e = \frac{2}{\pi} FL \quad (1.46)$$

essendo F la finezza ed L la lunghezza reale dei bracci. Il tempo di permanenza all'interno della cavità risulta

$$\tau = \frac{2}{\pi} \frac{FL}{c} \quad (1.47)$$

Quando questo tempo è superiore al periodo dell'onda gravitazionale la cavità media il segnale. A frequenza alte si ha dunque una diminuzione della sensibilità dell'interferometro ed il rumore *Shot* può scriversi

$$h(\nu) = \frac{\pi}{2FL} \sqrt{\frac{\hbar c \lambda}{4\pi P_{in}}} \sqrt{1 + \left(\frac{\nu}{\nu_c} \right)^2} \quad (1.48)$$

con $\nu_c = 500$ Hz.

Pressione di radiazione: aumentando la potenza del laser si potrebbe limitare lo shot noise. Tuttavia la luce esercita sugli specchi una forza proporzionale alla potenza stessa: $F_{rad} = P/c$. Questa forza fa oscillare gli specchi producendo uno spostamento

$$x(\nu) = \frac{1}{m(2\pi\nu)^2} F(\nu) \quad (1.49)$$

considerando la funzione di trasferimento lontana dal punto di risonanza. Poiché le fluttuazioni dei due bracci sono non correlate, si ricava l'ampiezza di densità spettrale:

$$h_{rad} = \frac{1}{m\nu^2 L} \sqrt{\frac{\hbar P_{in}}{2\pi^3 c \lambda}} \quad (1.50)$$

Rumore quantistico: a basse frequenze domina il rumore di pressione di radiazione, ad alte frequenze domina il rumore shot. Quando i due rumori si equivalgono vi è un minimo di densità spettrale totale. Questo si verifica solo se scegliamo la potenza di ingresso pari a

$$P(\nu) = \pi c \lambda m \nu^2 \quad (1.51)$$

In questo caso si raggiunge il limite quantistico

$$h(\nu) = \sqrt{\frac{\hbar}{\pi \nu L m}} \quad (1.52)$$

Solo scegliendo opportunamente la potenza del laser si può ottenere questo valore limite in sensibilità.

Rumore termico: tutte le componenti dell'interferometro operano a temperatura ambiente, per cui sono soggette al rumore termico. Per stimare l'influenza del rumore termico sul moto degli specchi si sfrutta il *teorema di fluttuazione e dissipazione*.

Supponiamo che il sistema sia soggetto ad una forza esterna $F_{ext}(\nu)$ alla quale risponde con una velocità $v(\nu)$. Nell'ipotesi di linearità del sistema, possiamo scrivere l'equazione del moto nella forma

$$F_{ext} = Z v(\nu) \quad (1.53)$$

oppure in modo equivalente

$$v(\nu) = Y F_{ext} \quad (1.54)$$

dove Z rappresenta l'impedenza e $Y = Z^{-1}$ l'ammettenza del sistema. Il teorema di fluttuazione e dissipazione afferma che lo spettro di potenza $F_{termico}^2(\nu)$ della forza fluttuante agente sul sistema è dato da

$$F_{termico}^2 = 4k_B T \Re(Z(\nu)) \quad (1.55)$$

dove T è la temperatura del sistema, k_B la costante di Boltzmann e $\Re(Z(\nu))$ è la parte reale dell'impedenza, quindi la parte dissipativa.

In forma più utile lo spettro di potenza del moto di fluttuazione del sistema può essere espresso da

$$x_{termico}^2 = \frac{k_b T}{\pi^2 \nu^2} \Re(Y(\nu)) \quad (1.56)$$

Questo teorema ci permette di avere una stima del rumore termico senza effettuare un'analisi microscopica, ma solo specificando un modello macroscopico dell'impedenza del sistema in funzione della frequenza. Così possiamo stimare il rumore termico dovuto ai modi di pendolo x_p , ai modi normali delle masse test x_s , ed ai modi di violino dei fili di sospensione x_v (figura 1.7).

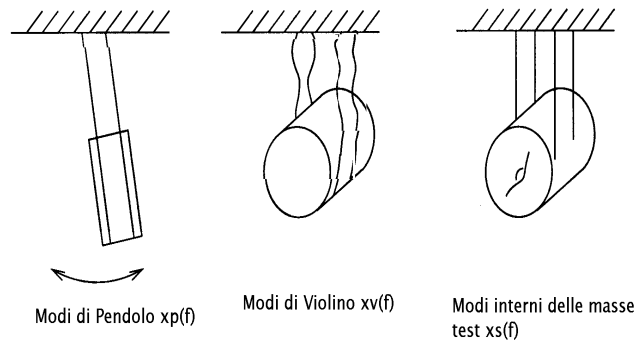


Figura 1.7: Effetti del rumore termico

Il contributo totale alla curva di sensibilità di VIRGO del rumore termico sarà

$$h_{termico}(\nu) = \frac{1}{L} \sqrt{x_p(\nu)^2 + x_v(\nu)^2 + x_s(\nu)^2} \quad (1.57)$$

- **Modi di pendolo:** consideriamo i due contributi principali: il moto flessurale di pendolo e la componente orizzontale del moto verticale dovuta alla curvatura della terra. Per la parte flessurale si ha

$$x(\nu)^2 = 4mk_B T \frac{(2\pi\nu_{0f})^2}{2\pi\nu} \phi(\nu) |H(\nu)|^2 \quad (1.58)$$

dove

$$H(\nu) = \frac{1}{m(2\pi\nu_0)^2} \frac{1}{1 - \left(\frac{\nu}{\nu_0}\right)^2 + i \left(\frac{\nu}{\nu_{0f}}\right)^2} \phi(\nu) \quad (1.59)$$

è la funzione di trasferimento del moto flessurale. La funzione $\phi(\nu)$ è detta angolo di perdita dei fili ed è la parte dissipativa del sistema "filo di sospensione". La frequenza di risonanza ν_{0f} in assenza di gravità per gli specchi vicini e quelli lontani è ovviamente diversa perchè gli specchi hanno masse diverse. In presenza di gravità la frequenza di risonanza è

$$\nu_0 = \frac{1}{2\pi} \sqrt{\frac{g}{L_f} + (2\pi\nu_{0f})^2} \quad (1.60)$$

con L_f pari alla lunghezza dei fili. Per la componente orizzontale abbiamo

$$x(\nu)^2 = 4mk_B T \frac{(2\pi\nu_{0v})^2}{2\pi\nu} \phi(\nu) |H(\nu)|^2 \theta^2 \quad (1.61)$$

dove $\theta = 2.35 \cdot 10^{-4} \text{ rad}$ è l'angolo di curvatura locale e

$$H(\nu) = \frac{1}{m(2\pi\nu_{0v})^2} \frac{1}{1 - \left(\frac{\nu}{\nu_{0v}}\right)^2 + i\phi(\nu)} \quad (1.62)$$

è la funzione di trasferimento del modo di pendolo verticale con frequenza di risonanza ν_0 . Il contributo totale è dato da due volte, a causa dei due bracci dell'interferometro, la somma dei due spettri di potenza.

- **Modi degli specchi:** lo spostamento longitudinale dovuto ai modi normali delle masse test è calcolato come la somma di due contributi: il contributo dei due modi di tamburo degli specchi, detti *modi di drum*, sotto i 10^4 Hz, e la coda a bassa frequenza di tutti i modi ad alta frequenza.

Per il primo abbiamo

$$x(\nu)^2 = 4k_B T \frac{1}{m_{ms}} \frac{\phi_q}{(2\pi\nu)(2\pi\nu_{ms})^2 \left(1 - \left(\frac{\nu}{\nu_{ms}}\right)^2\right)^2 + \phi^2} \quad (1.63)$$

dove $\phi_q = 10^{-6}$ e ν_{ms} è la frequenza di risonanza dei modi drum.

per il secondo abbiamo

$$x(\nu)^2 = 4k_B T \frac{1}{2\pi\nu} \phi_q m_w \quad (1.64)$$

dove $m_w = ((m_{ms}\omega^2))^{-1}$ è il parametro di massa equivalente sommato su tutti i modi.

- **Modi di violino:** ciascun filo di sospensione ha una propria frequenza di risonanza che può essere eccitata dal rumore termico. Per stimare il contributo dei modi di violino associamo ciascuna risonanza del filo ad un oscillatore con frequenza di risonanza pari a:

$$\nu_n = \frac{n}{2L_f} \sqrt{\frac{mg}{4\rho_f \pi r^2}} \left(1 + \frac{2}{L_f} \sqrt{\frac{4EI_2}{mg}}\right) \quad (1.65)$$

Il contributo totale si ottiene sommando su tutti i modi di violino e tenendo conto che ciascuno specchio è sorretto da 4 fili.

$$x_v(\nu)^2 = 4 \cdot 4mk_B T \frac{2\rho_f r^2 L_f}{\pi m} \sum_n \frac{\phi_{vn}}{(2\pi\nu)(2\pi\nu_n)^2 n^2 \left(\left(1 - \left(\frac{\nu}{\nu_n}\right)^2\right)^2 + \phi_{vn}^2\right)} \quad (1.66)$$

dove

$$\phi_{vn} = \frac{2\phi(\nu_n)}{L_f} \sqrt{\frac{4EI_2}{mg}} \left(1 + \frac{(n\phi(\nu_n))^2}{2L_f} \sqrt{\frac{4EI_2}{mg}}\right)^{-1} \quad (1.67)$$

è l'angolo di perdita associato ai modi di violino, ρ_f la densità di massa dei fili, r il loro raggio, E il modulo di Young e I_2 il momento di inerzia del filo.

Rumore newtoniano: le fluttuazioni di densità di massa prodotte dalle onde di microsisma producono un campo gravitazionale stocastico che si accoppia direttamente alle masse test dell'interferometro causando il *rumore newtoniano*. Nell'intervallo di frequenze in cui il rumore newtoniano è rilevante, possiamo descrivere la base del superattenuatore come un corpo rigido con tre modi di vibrazione, due orizzontali ed uno verticale, e tre modi di rotazione, uno intorno all'asse del superattenuatore e due di *tilt*.

L'effetto di una fluttuazione di densità di massa

$$\delta\rho(t) = \rho(t) - \langle \rho(t) \rangle \quad (1.68)$$

posta in \mathbf{R} rispetto ad una massa test è uno spostamento \mathbf{x} che può essere scritto nello spazio delle frequenze come ⁴

$$-\omega^2 \mathbf{x}(\omega) = \frac{\mathbf{F}}{m} = G\delta\rho(\omega) \frac{\mathbf{R}}{R^3} dV \quad (1.69)$$

Abbiamo supposto che le masse test fossero tre, compresa una unica massa che funge da *beam splitter* sospesa all'angolo della L tra le due masse test poste alla fine dei bracci. In questa approssimazione, la differenza di cammini ottici è

$$\begin{aligned} -\omega^2 \delta L(\omega) = & G\delta\rho(\omega) \left\{ \frac{(L-y)}{(x^2 + (L-y)^2 + z^2)^{3/2}} - \frac{(-y)}{(x^2 + y^2 + z^2)^{3/2}} \right. \\ & \left. - \frac{(L-x)}{((L-x)^2 + y^2 + z^2)^{3/2}} + \frac{(-x)}{(x^2 + y^2 + z^2)^{3/2}} \right\} dV \end{aligned} \quad (1.70)$$

essendo L la lunghezza dei due bracci dell'interferometro. In questa descrizione è contenuta la struttura geometrica dell'apparato.

⁴Lontani dalle frequenza di risonanza dei pendoli, la funzione di trasferimento è proporzionale a ω^2

Dato allora un modello di dinamica del terreno, possiamo integrare l'espressione 1.70 su tutte le fluttuazioni di massa ottenendo una relazione fra lo spettro di rumore sismico, che può essere misurato, e quello newtoniano. In questo lavoro non entreremo in dettaglio nella stima del rumore newtoniano. Per ulteriori approfondimenti si può vedere [3].

Curva di sensibilità di VIRGO

A questo punto possiamo stimare la curva di sensibilità dell'interferometro. Il suo spettro di rumore, che riportiamo in figura 1.8, è dato dalla sovrapposizione dei contributi analizzati nei paragrafi precedenti. In aggiunta si notano due risonanze, intorno ai 6-8 Hz, chiamate *creep noise*, dovute ai microcedimenti dei fili di sospensione degli specchi. La parte a bassa frequenza è dominata principalmente dal rumore termico, quella ad alte frequenze dal rumore shoot. La migliore sensibilità si ha fra i 10 e 10000 Hz.

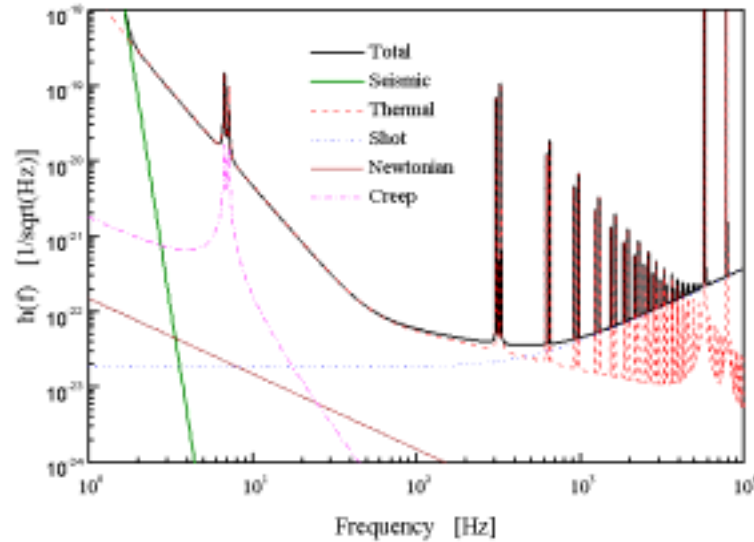


Figura 1.8: Curva di sensibilità del rivelatore VIRGO

Bibliografia relativa al capitolo I

- [1] P.G. Bergmann, *Introduction to the Theory of Relativity*, Dover Publication, New York, 1976
- [2] P.R. Saulson, *Interferometric Gravitational Wave Detection*, Word Scientific, 1994
- [3] E. Cuoco, *Identificazione dello spettro di rumore di VIRGO e metodi di rivelazione di onde gravitazionali*, Tesi di Dottorato in Fisica, Univesità degli studi di Pisa, 1997
- [4] L.S. Finn, *Detection, measurement, and gravitational radiation*, Phy. Rev. D, 46(12):5236-5249, 1992

Capitolo 2

Analisi di segnali nel rumore

Le onde gravitazionali causano un effetto molto debole sull'apparato di misura. Così è naturale riflettere sul modo in cui possiamo registrare questi effetti, in presenza di un certo livello non noto di rumore. Discussioni qualitative sono basate sul concetto di rapporto segnale rumore. Una misura accurata può essere effettuata quando l'effetto principale delle variazioni dell'uscita del nostro strumento (ciò che noi chiamiamo *segnale*) è grande rispetto alle variazioni casuali dell'output quando il segnale non è presente (ciò che noi chiamiamo *rumore*).

Vedremo in questo capitolo le principali misure che possono essere effettuate su un segnale, sia quando è generato da un sistema lineare che quando diventano predominanti gli effetti non lineari, in modo da rendere quantitative le nostre misure.

2.1 Caratterizzazione di una serie temporale

Nella ricerca delle onde gravitazionali principalmente analizziamo l'uscita, o quando è possibile l'ingresso, del nostro rivelatore in funzione del tempo, cioè una *serie temporale*. Possiamo immaginare due possibili serie temporali: una deterministica con una esplicita forma matematica ed una stocastica, che varia da una realizzazione

all'altra. Il motivo di questa distinzione è che utilizziamo diversi strumenti per caratterizzare la presenza di regolarità in una serie deterministica o stocastica.

2.1.1 Trasformata di Fourier

Consideriamo una funzione deterministica del tempo $s(t)$. I valori di questa funzione per un insieme di istanti di tempo diversi la definiscono univocamente. Un modo alternativo per avere le stesse informazioni è la *trasformata di Fourier* (FT) di $s(t)$, denominata $S(f)$, data da:

$$S(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) e^{-i2\pi ft} dt \quad (2.1)$$

o dalla forma discreta (DFT: discrete Fourier trasform) se usiamo dati campionati:

$$x[f] = \sum_{n=-\infty}^{+\infty} x[n] \exp(-i2\pi fn) \quad (2.2)$$

Vedremo fra breve i vantaggi dell'analisi nel dominio delle frequenze.

2.1.2 Cross-correlazione e autocorrelazione

Consideriamo ora un modo per caratterizzare una serie stocastica nel dominio delle frequenze. Dobbiamo prima descrivere due operazioni che possono essere effettuate fra due funzioni $s_1(t)$ e $s_2(t)$.

Definiamo *correlazione* (cross-correlation) fra le due funzioni suddette la quantità:

$$(s_1 \star s_2)(\tau) \equiv \int_{-\infty}^{+\infty} s_1(t) s_2(t + \tau) dt \quad (2.3)$$

La correlazione è un modo per misurare il grado di similitudine di due funzioni.

Quando le funzioni $s_1(t)$ e $s_2(t)$ sono la stessa funzione $s(t)$ abbiamo l'*autocorrelazione*:

$$(s \star s)(\tau) \equiv \int_{-\infty}^{+\infty} s(t) s(t + \tau) dt \quad (2.4)$$

Questo è un modo per considerare quanto una funzione è uguale ad una sua replica sfasata. Si ha un massimo per $\tau = 0$ e le funzioni periodiche hanno massimi multipli del periodo. La funzione di autocorrelazione per $\tau = 0$ rappresenta il valor medio della funzione e per serie temporali a media nulla, $s \star s(\tau = 0)$ è la deviazione standard. In altre parole, una misura del rumore. Principalmente $s \star s$ indica quanto velocemente varia la funzione $s(t)$ nel tempo. In genere si normalizza per $s \star s(\tau = 0)$ in modo da avere il valore massimo pari all'unità per $\tau = 0$.

2.1.3 Convoluzione

Un'altra operazione molto importante è la *convoluzione* fra due funzioni definita da:

$$(s_1 * s_2)(\tau) \equiv \int_{-\infty}^{+\infty} s_1(t) s_2(t - \tau) dt \quad (2.5)$$

L'importanza della convoluzione compare principalmente nell'analisi dei sistemi. Ricordiamo infatti che la risposta di un sistema ad una sollecitazione esterna si calcola effettuando la convoluzione fra il segnale in ingresso e la risposta all'impulso della funzione di trasferimento del sistema stesso.

È possibile applicare tutte queste operazioni su ogni tipo di funzione matematica, sia essa deterministica o una particolare realizzazione di un processo stocastico. In genere si pone particolare attenzione sull'operazione che più chiaramente caratterizza la presenza di regolarità.

2.1.4 Spettro di potenza

La trasformata di Fourier $S(f)$ è un ottimo modo per osservare una serie temporale deterministica nel dominio delle frequenze. Se $s(t)$ è una serie generata da un processo stocastico è più utile calcolare lo *spettro di potenza* definito come la trasformata di Fourier della funzione di autocorrelazione:

$$P_s(f) \equiv \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s \star s(\tau) e^{-i2\pi f\tau} d\tau \quad (2.6)$$

Questa trasformazione contiene tutte le informazioni insite nella funzione di autocorrelazione: $P_s(f)$ è una misura della variazione della serie temporale alla frequenza f .

Spesso è conveniente ragionare in termini di seni e coseni di frequenze positive, anziché di esponenziali complessi. Definiamo allora lo spettro di potenza *single-side*:

$$s^2(f) \equiv \begin{cases} 2P_s(f), & \text{se } f \geq 0 \\ 0, & \text{altrimenti} \end{cases} \quad (2.7)$$

In tutto il lavoro di tesi faremo sempre riferimento alla versione *single-side* dello spettro. Il valor medio di tale funzione è

$$\langle s^2 \rangle = \int_0^{+\infty} s^2(f) df \quad (2.8)$$

e rappresenta la pototenza del segnale.

2.2 Rapporto segnale-rumore

Abbiamo ora sviluppato le basi per introdurre il concetto di *rapporto segnale-rumore* o *SNR*. La difficoltà emerge dalla diversa dimensione delle unità di misura usate: per esempio, se il segnale è misurato in volts, o in volt/Hz nello spazio di Fourier, un processo stocastico ha lo spettro di potenza misurato in volt^2/Hz o densità spettrale in $\text{volt}/\sqrt{\text{Hz}}$. Discuteremo la questione prima nel dominio temporale, e successivamente nel dominio delle frequenze.

2.2.1 Statistica del rumore

Consideriamo un rivelatore di rumore ed al suo output $v(t)$. Quando il segnale non è presente l'output varia in modo casuale¹.

Per poter distinguere il segnale dal rumore, dobbiamo assumere che ne conosciamo una particolare forma, come funzione del tempo $s(t)$, che chiamiamo *template*.

¹Assumiamo, senza perdere di generalità, che abbia media nulla e varianza unitaria

Signal detection è il processo secondo il quale si cerca di distinguere le variazioni di potenza del rumore con aggiunto il segnale, rispetto a quando è presente solo il rumore.

Quando analizziamo una registrazione è semplice cercare un accordo fra questa ed il template: calcolando la correlazione

$$v \star s(t) = \int_{-\infty}^{+\infty} v(t)s(t+\tau)d\tau \quad (2.9)$$

è possibile una distinzione nel tempo del momento in cui è arrivato il segnale.

Consideriamo una serie temporale senza rumore. Chiaramente quando il template $s(t)$ coincide con il segnale $v(t) = \alpha s(t-t_0)$ registrato, la correlazione è massima e vale $\int_{\mathbb{R}} \alpha s^2(\tau)d\tau$. Se il template è allineato con una porzione di registrazione in cui l'output è zero (cioè non è presente né il segnale né il rumore), la cross-correlazione vale zero.

Se il rumore all'uscita del rivelatore ha distribuzione di probabilità gaussiana:

$$P(v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-v^2/2\sigma^2\} \quad (2.10)$$

allora, dato l'esiguo rapporto segnale-rumore aspettato, $v \star s(t)$ ha una distribuzione di probabilità anch'essa gaussiana.

Può accadere che, se abbiamo un segnale presente particolarmente ampio, l'output del rivelatore è molto simile al template ed il valore di $v \star s$ aumenta. Sull'istogramma di $P(v)$ noteremo una variazione della distribuzione degli eventi, rispetto a quando era presente solo il rumore. Questo "outlier" è un indice della presenza del segnale.

Possiamo ora definire in modo univoco il rapporto segnale-rumore. Per ogni template, caratterizziamo il rumore dalla quantità:

$$N^2 \equiv \sqrt{\langle (v \star s(\tau))^2 \rangle} \quad (2.11)$$

dove il simbolo $\langle \cdot \rangle$ indica la media temporale. Questa è una misura della larghezza della funzione di distribuzione. Possiamo poi caratterizzare l'intensità S^2 del segnale presente al tempo t dalla correlazione fra il template e l'output

$$S^2 \equiv \|v \star s(t)\| \quad (2.12)$$

Il rapporto segnale-rumore, o SNR, è la radice quadrata del rapporto fra questa misura della quantità di segnale presente, ed il valore atteso dovuto al solo rumore:

$$SNR \equiv \sqrt{S^2/N^2} \quad (2.13)$$

abbiamo così normalizzato il segnale al rumore.

Il SNR è una misura monotona dell'improbabilità di osservare un dato valore della correlazione, con l'assunzione che l'output contiene solo rumore, senza segnale. Così un alto valore di SNR è indicativo della presenza di qualcosa di diverso dal rumore del rivelatore.

2.2.2 Matched filters

Nel paragrafo precedente abbiamo visto che il modo migliore per rivelare il segnale è calcolare la correlazione fra l'output rumoroso del rivelatore ed un template. Questa è la scelta migliore quando conosciamo il template e quando il rumore del rivelato è gaussiano e bianco, cioè con ampiezza spettrale indipendente dalla frequenza. Indichiamo questa procedura con il termine *Matched Template*. Il procedimento è leggermente più complicato se il rumore non è bianco, come vedremo in seguito.

Analizziamo prima un caso realistico: vogliamo rivelare la presenza del segnale registrando serie temporali non troppo lunghe, cioè realizzare un apparato che effettui l'operazione di correlazione in tempo reale. La risposta è nella somiglianza fra la correlazione e l'integrale di convoluzione. Ricordiamo che la sola differenza è nel segno meno presente nell'operazione di convoluzione.

Questa complicazione può essere completamente risolta se noi realizziamo un sistema lineare, detto *Matched filter*, la cui risposta all'impulso è l'inversione temporale del segnale $s(t)$ che stiamo cercando. Se applichiamo l'output rumoroso del nostro rivelatore al matched filter, allora rivelare grandi valori nell'output del filtro equivale a trovare inaspettate variazioni nella correlazione con il template.

Possiamo descrivere il processo nel dominio delle frequenze. La risposta all'impulso del matched filter è $g(\tau) = s(-\tau)$ ed ha trasformata di Fourier:

$$G(f) = S^*(f) \exp(-i2\pi f t_0) \quad (2.14)$$

dove t_0 è il parametro che descrive la durata della serie temporale usata dal filtro per calcolare la cross-correlazione. L'output propriamente filtrato ha la forma $x(f) \propto G(f)V(f)$. La proporzionalità a $S^*(f)$ mostra che il filtro lascia passare le frequenze in cui c'è maggiore presenza del segnale.

Possiamo ora considerare il caso in cui il rumore non è bianco. L'idea di base è che devono essere enfatizzate le frequenze in cui il rapporto fra l'ampiezza del segnale e la potenza del rumore è più grande, escludendo le frequenze in cui il SNR è piccolo. Allora il filtro ottimale ha funzione di trasferimento:

$$G(f) \propto \frac{S^*(f) \exp(-i2\pi f t_0)}{N^2(f)} \quad (2.15)$$

Possiamo infine pensare in modo euristico ad un filtraggio in due fasi: prima si filtra il segnale con il denominatore del matched filter, in modo da rendere il rumore bianco, poi si usa il filtro definito al numeratore.

In campi di ricerca come l'astrofisica delle onde gravitazionali, allo stato attuale, si ritiene di conoscere l'esatta forma del segnale che giunge al nostro rivelatore, si conoscono le frequenze caratteristiche che ben lo rappresentano ma non sono noti i parametri della forma d'onda. Con queste conoscenze possiamo realizzare un filtro ottimo che lascia passare solo le bande di frequenza in cui cerchiamo il segnale, filtrando il rumore con frequenze esterne a tali bande.

Ci chiediamo ora qual è il SNR quando usiamo un filtro per cercare un particolare segnale. Indichiamo con s_0 l'ampiezza caratteristica del segnale. Il rapporto segnale-rumore è

$$SNR \sim \frac{s_0}{\sqrt{n^2(f)\Delta f}} \quad (2.16)$$

dove $n(f)$ è lo spettro di potenza del rumore e Δf è l'ampiezza della banda passante del filtro.

Ora è immediatamente chiara l'importanza di riuscire a mediare un segnale per lungo tempo. Per esempio, se un segnale impulsivo, generato dal collasso di una supernova, è presente per 1 ms, l'ampiezza di banda è 1 KHz. Il valore numerico del *rms* del rumore è $\sqrt{\Delta f}$ volte il valore dell'ampiezza della sua densità spettrale. Per un segnale di 1 ms in un rumore di $1 \mu V/\sqrt{Hz}$ abbiamo un rms di circa $30 \mu V$. D'altra parte se abbiamo un segnale che oscilla per 100 sec, ci aspettiamo di riuscire a rivelare ampiezze più piccole. Il rms del rumore che passa il matched filter è circa $1 \mu V/\sqrt{Hz}$ diviso la radice quadrata di 100 sec, cioè $0.1 \mu V$.

Ricordiamo infine che quando paragoniamo la rilevanza di due segnali differenti, valutiamo le ampiezze dei picchi. Ma quando consideriamo le possibilità di rivelazione dobbiamo includere i benefici del mediare i segnali, misurando il numero di volte N che si manifesta la massima ampiezza. È utile in tal caso paragonare la rilevanza di due segnali non valutando s_0 , ma l'ampiezza caratteristica $s_c \equiv s_0\sqrt{N}$.

2.3 Dati caotici in sistemi fisici

In questa sezione descriveremo nuove tecniche per quantificare l'osservazione di sistemi fisici in cui la traccia temporale delle variabili dinamiche sono irregolari o caotiche, la cui interpretazione, con i metodi standard di analisi, è semplicemente di classificare il segnale come irregolare.

Molti dei lavori sulla dinamica non lineare riportati in passato si sono limitati ad

analizzare il sistema in approssimazione lineare con uno sviluppo in serie al primo ordine. Solo recentemente si è mostrato interesse alle misure sulle serie caotiche, in modo da coglierne proprietà che ne permettessero la predizione, il filtraggio e l'identificazione in un contesto stocastico [1], [4].

2.3.1 Sistemi dinamici e Caos

L'apparenza di una dinamica non regolare è ristretta ai sistemi non lineari. I sistemi lineari godono della proprietà che ad una combinazione lineare degli ingressi corrisponde una combinazione lineare delle uscite. Dunque una variazione delle condizioni iniziali o dei parametri non comporta una variazione qualitativa dell'evoluzione². In un sistema autonomo (non dipendente esplicitamente dal tempo) lineare con f gradi di libertà, $\mathbf{u}(t) = [u_1(t), \dots, u_f(t)]$, l'equazione di evoluzione è

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{A} \cdot \mathbf{u}(t) \quad (2.17)$$

dove \mathbf{A} è una matrice costante $f \times f$. La soluzione di questa equazione lineare tende al punto $\mathbf{u}(\infty) = \mathbf{0}$ se tutti gli autovalori della matrice \mathbf{A} hanno parte reale negativa. Se qualche autovalore ha parte reale positiva, allora la soluzione lungo il sottospazio individuato dall'autovettore corrispondente diverge [6]. Data la struttura del set di equazioni differenziali descriventi un sistema lineare, notiamo che l'evoluzione è perfettamente prevedibile: una piccola variazione delle condizioni iniziali, che possono generarsi per effetto degli errori di misura, porta ad andamenti asintotici simili, all'interno degli stessi errori di misura.

In generale, nell'analisi di sistemi fisici, assumiamo che, per dinamiche a tempo continuo esiste un set di f equazioni differenziali ordinarie nelle variabili $\mathbf{u}(t)$,

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{G}(\mathbf{u}(t)) \quad (2.18)$$

²Per esempio, un oscillatore armonico ha un andamento periodico indipendentemente dalla costante elastica k e dalle condizioni iniziali, anche se variano l'ampiezza e la frequenza.

dove il campo vettoriale $\mathbf{G}(\mathbf{u}(t))$ è considerato almeno continuo a tratti e differenziabile. Quando il tempo è discreto ³, l'evoluzione è data da una mappa di vettori in \mathfrak{R}^f ad altri vettori in \mathfrak{R}^f , ognuno etichettato dal tempo discreto $\mathbf{u}(n) = \mathbf{u}(t_0 + n\tau_s)$, avendo indicato t_0 e τ_s rispettivamente il tempo iniziale e l'intervallo di campionamento. L'evoluzione della variabile discreta è regolata dalla equazione:

$$\mathbf{u}(n+1) = \mathbf{F}(\mathbf{u}(n)) \quad (2.19)$$

I modelli di dinamica a tempo continuo o a tempo discreto possono essere connessi pensando di approssimare la derivata temporale con

$$\frac{d\mathbf{u}(t)}{dt} \approx \frac{\mathbf{u}(t_0 + (n+1)\tau_s) - \mathbf{u}(t_0 + n\tau_s)}{\tau_s} \quad (2.20)$$

che può essere trasformato in

$$\mathbf{F}(\mathbf{u}(n)) \approx \mathbf{u}(n) + \tau_s \mathbf{G}(\mathbf{u}(n)) \quad (2.21)$$

come relazione fra modelli a tempo continuo o a tempo discreto.

Il numero di gradi di libertà o, equivalentemente, la dimensione dello spazio degli stati di un sistema dinamico è uguale al numero di equazioni differenziali del primo ordine necessarie per descrivere l'evoluzione. Il campo vettoriale $\mathbf{F}(\mathbf{u})$ ha parametri che riflettono le proprietà fisiche del sistema come forzanti, effetti dissipativi ecc. Nel caso di un circuito non lineare, per esempio, questi parametri riflettono ampiezze e frequenze caratteristiche dei dispositivi di corrente, valori di resistenze, capacità e qualsiasi caratteristica corrente-tensione di elementi non lineari.

Il sistema dinamico $\mathbf{u}(n+1) = \mathbf{F}(\mathbf{u}(n))$ non conserva generalmente il volume nello spazio delle fasi f -dimensionale. Nell'evoluzione da un volume $d^f \mathbf{u}(n)$ ad un

³In effetti, dal punto di vista sperimentale, si è sempre in questa situazione in quanto ogni strumento di misura effettua un campionamento che non può andare al di sotto del tempo morto tipico dello strumento.

volume $d^f \mathbf{u}(n+1)$, si ha una variazione dipendente dallo Jacobiano

$$J(\mathbf{F}(\mathbf{u})) = \det \left(\frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}} \right) \quad (2.22)$$

Quando lo Jacobiano è unitario, che è un caso speciale associato alla evoluzione della Hamiltoniana dipendente dai parametri del sistema, allora il volume nello spazio delle fasi rimane costante. Generalmente lo Jacobiano, in molti sistemi fisici, è minore dell'unità e quindi il volume si contrae nell'evoluzione e si manifestano principalmente effetti dissipativi.

In generale, gli insiemi su cui si svolge la dinamica per $t \rightarrow \infty$ sono detti **attrattori**. Al variare dei parametri si possono avere principalmente tre comportamenti possibili: punti di equilibrio, cicli limite, soluzioni quasi periodiche. All'attrattore è associata una dimensione che è caratteristica del sistema fisico analizzato.

Quando i parametri del sistema variano, indipendentemente dalle condizioni iniziali $\mathbf{u}(t_0)$, per t grandi ci aspettiamo che vari anche $\mathbf{u}(t)$. Alcune volte i cambiamenti sono regolari e si conserva la geometria di un set di punti in \mathbb{R}^f . Altre volte, piccole variazioni dei parametri, o piccole variazioni delle condizioni iniziali, porta ad una evoluzione del sistema del tutto inaspettata, e può non conservarsi la geometria stessa del volume nello spazio delle fasi. In questo caso si dice che il sistema è caotico.

Un definizione più rigorosa è:

Il moto di un sistema confinato in una regione limitata dello spazio delle fasi si dice **caotico** se verifica una equazione differenziale ordinaria o alle derivate parziali (quindi è una evoluzione deterministica) e se esiste almeno un osservabile che sia aperiodico o che abbia un numero di frequenze superiori al numero di gradi di libertà del sistema

Osserviamo che tutti i sistemi caotici sono non lineari ma non vale il contrario. Notiamo inoltre che, poichè il sistema è deterministico, utilizzando la stessa con-

dizione iniziale, si ha la medesima evoluzione, a differenza dei sistemi stocastici, il cui medesimo valore iniziale porta ad evoluzioni diverse compatibilmente alla funzione di distribuzione di probabilità che caratterizza la dinamica.

Per avere un esempio esplicito consideriamo il sistema di equazioni differenziali di Lorenz:

$$\begin{aligned}\frac{dx(t)}{dt} &= \sigma(y(t) - x(t)) \\ \frac{dy(t)}{dt} &= -x(t)z(t) + rx(t) - y(t) \\ \frac{dz(t)}{dt} &= x(t)y(t) - bz(t)\end{aligned}\tag{2.23}$$

che derivano da una semplificazione del sistema di equazioni differenziali alle derivate parziali che descrivono i moti convettivi termici nella bassa atmosfera.

Il parametro più interessante dal punto di vista fisico è r che rappresenta l'influenza delle forze esterne sull'atmosfera. Il valore per cui si ha la migliore rappresentazione dei moti termici convettivi è per $r \approx 1$. Per $r < 1$ tutte le orbite tendono ad un punto fisso $(x, y, z) = (0, 0, 0)$, cioè conduzione termica costante. Quando r supera l'unità lo stato $(0, 0, 0)$ è linearmente instabile ed il campo vettoriale ha due punti simmetrici linearmente stabili a $(x_{\pm}, y_{\pm}, z) = (\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1)$. Per $1 < r < r_c$ con $r_c = \sigma(\sigma + b + 3)/(\sigma - b - 1)$, la geometria dello stato finale dell'orbita nello spazio delle fasi è la stessa, indipendentemente dallo stato iniziale, lo stato finale è sempre un punto.

Per $r > r_c$ la situazione cambia drammaticamente: i due punti fissi del campo vettoriale diventano linearmente instabili e non esiste nessun punto fisso stabile, il moto è irregolare ed è caratterizzato da uno spettro di Fourier a larga banda, senza alcuna struttura periodica. La complessa evoluzione manifestata nella serie temporale è dovuta al fatto che il punto, nello spazio delle fasi, percorre un'orbita non chiusa, ma limitata: questo oggetto geometrico, la cui dimensione è frazionaria, si chiama *attrattore strano*. Notiamo subito che tutti i sistemi caotici possiedono un

attrattore strano. Non è vero invece il contrario: esistono sistemi, come gli oscillatori non lineari con due forzanti a frequenze incommensurabili, che hanno un attrattore strano, ma non sono caotici.

In figura 2.1 vediamo le tre componenti come risulta risolvendo le equazioni di Lorenz con il metodo di Runge-Kutta del quarto ordine. I valori dei parametri sono $r = 45.92$, $b = 4.0$ e $\sigma = 16.0$.

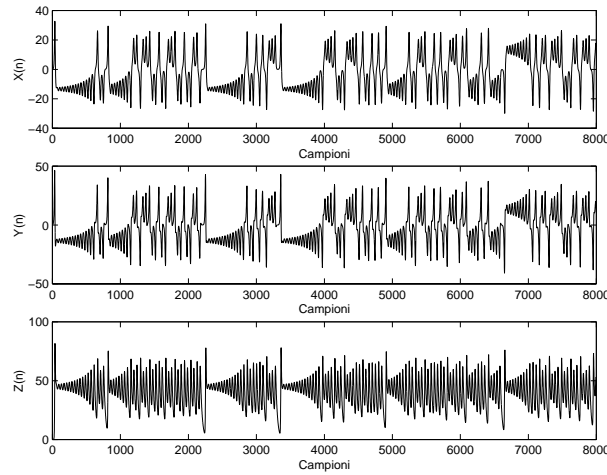


Figura 2.1: Evoluzione temporale delle tre componenti del sistema di Lorenz

In figura 2.2 si nota l'orbita dei tre gradi di libertà $[x(t), y(t), z(t)]$ nello spazio tridimensionale. Si vede chiaramente la struttura dell'attrattore strano di Lorenz.

Lo spettro a larga banda è il primo indicatore di un comportamento caotico, anche se non è sufficiente per classificare il sistema come caotico. In figura 2.3 si può notare come lo spettro di potenza del sistema di Lorenz decresca come una potenza.

L'analisi lineare non dà risultati quando si lavora con sistemi con spettri di questo tipo. I modelli autoregressivi a media mobile (ARMA) non sono utili per analizzare questi sistemi. Se si prova ad interpolare i dati generati da una sorgente caotica con un modello lineare, usando i minimi quadrati per trovare i parametri, si nota che

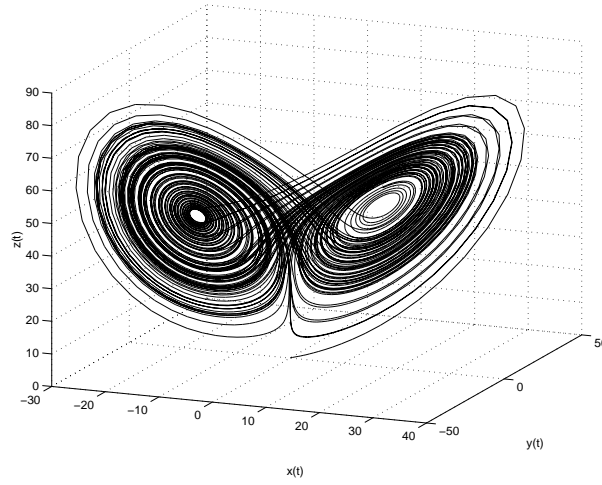


Figura 2.2: Evoluzione 3D del sistema di Lorenz

il valore del *RMS* è dello stesso ordine di grandezza dell'attrattore, e quindi non è stata identificata la dinamica del sistema.

Se si prova un fit locale lineare fra i punti vicini nello spazio delle fasi, che è il modello non lineare più semplice, l'errore quadratico medio può essere ridotto, ma dipende molto dal numero di dati a disposizione e dalla distanza fra i punti vicini.

2.3.2 Ricostruzione dello spazio delle fasi

Consideriamo il metodo per ricostruire lo spazio delle fasi di un sistema quando abbiamo informazione su di una misura scalare $s(n) = s(t_0 + n\tau_s)$. Dalle idee sviluppate precedentemente sui sistemi dinamici abbiamo visto che, se riusciamo a valutare la derivata prima di una variabile misurata, possiamo immaginare di trovare la connessione tra le varie derivate e le variabili di stato, cioè l'equazione differenziale che genera l'osservazione. Per trovare il valore

$$\dot{s}(n) = \left. \frac{ds(t)}{dt} \right|_{t=t_0+n\tau_s} \quad (2.24)$$

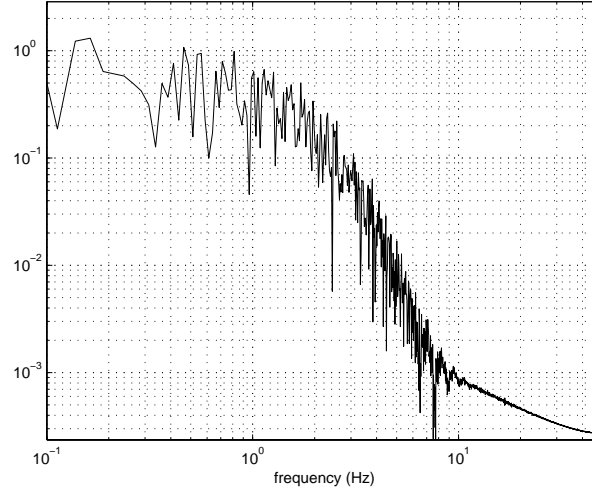


Figura 2.3: Spettro di potenza della componente x del sistema di Lorenz

dobbiamo approssimare la derivata con

$$\dot{s}(n) \approx \frac{s(t_0 + (n+1)\tau_s) - s(t_0 + n\tau_s)}{\tau_s} \quad (2.25)$$

Con un τ_s finito, questo è un filtro passa alto, la rappresentazione del sistema dinamico è scadente e peggiora passando alle derivate di ordine superiore.

Nel 1980 circa due gruppi indipendenti, il gruppo di Santa Cruz (Packard e altri) e David Ruelle con Mañé, introdussero l'uso delle coordinate ritardate per ricostruire lo spazio delle fasi di un sistema dinamico. L'idea principale è che in realtà non abbiamo bisogno delle derivate per formare un sistema di coordinate nel quale catturare l'evoluzione del sistema nello spazio delle fasi, ma che possiamo direttamente usare le variabili ritardate $s(n + jT)$ con $i = 0, \dots, d_E - 1$, dove T e d_E sono due interi da determinare. Allora, usando un vettore di campioni ritardati della serie

$$\mathbf{y}(n) = [s(n), s(n+T), s(n+2T), \dots, s(n+(d_E-1)T)] \quad (2.26)$$

abbiamo il sistema di coordinate necessario. In un sistema dinamico non lineare le $s(n + jT)$ sono combinazioni non lineari sconosciute delle variabili fisiche originarie.

Come possiamo scegliere ora il valore di T , detto *time lag*, e di d_E , cioè il numero giusto delle coordinate? Takens (1981) e Ruelle suggeriscono che non è importante il time lag che si sceglie, ed il numero di coordinate dipende dalla dimensione dell'attrattore. Essi affermano che se d , che è un intero, è più grande di $2d_A$, con d_A pari alla dimensione dell'attrattore, allora l'attrattore che si vede nello spazio delle coordinate ritardate è una rappresentazione smooth dell'attrattore che si vede nello spazio delle fasi originario, che a priori non conosciamo. In pratica il loro teorema dà come condizione sufficiente lo scegliere $d_E = \min(d > 2d_A)$ con $d \in \mathbb{N}$ per riuscire a non perdere la struttura geometrica originale dell'attrattore, quando passiamo dalla rappresentazione nello spazio delle fasi, che non conosciamo, a quella nello spazio delle coordinate ritardate. La procedura per scegliere d sufficientemente grande è detta *embedding*, e la dimensione trovata d_E è detta *dimensione di embedding*. Da notare che, una volta scelto $d = d_E$, per qualsiasi $d > d_E$ la struttura dell'attrattore non cambia. Sono stati proposti anche altri metodi per ricostruire lo spazio degli stati, ma il *time delay embeddin*, è il solo metodo sistematico e abbastanza semplice da implementare per passare da una serie scalare allo spazio delle fasi multidimensionale [2].

Scelta del time delay

Parte del teorema di embedding afferma che qualsiasi time lag è accettabile e non è molto importante ai fini dell'informazione fisica insita nella serie temporale. Tuttavia se scegliamo T troppo piccolo, allora, per un campionamento fitto dai dati, la coordinata $s(n + jT)$ e la coordinata $s(n + (j + 1)T)$ differiscono poco numericamente e non sono indipendenti. Similmente, se scegliamo T troppo grande, allora $s(n + jT)$ e $s(n + (j + 1)T)$ sono completamente indipendenti l'una dall'altra dal punto di vista statistico e la proiezione dell'orbita dell'attrattore avviene in due direzioni completamente non correlate. L'origine di questa indipendenza statistica è

l'ambigua instabilità dei sistemi caotici.

Una prima idea è considerare $s(n)$ come generata da una certa distribuzione non nota. Allora, calcolando la funzione di autocorrelazione lineare (ACF)

$$C_L(\tau) = \frac{\frac{1}{N} \sum_{m=1}^N [s(m+\tau) - \bar{s}][s(m) - \bar{s}]}{\frac{1}{N} \sum_{m=1}^N [s(m) - \bar{s}]^2} \quad (2.27)$$

dove

$$\bar{s} = \frac{1}{N} \sum_{m=1}^N s(m) \quad (2.28)$$

possiamo rappresentare $C_L(\tau)$ in funzione di τ . Se l'ACF decresce lentamente, vuol dire che la serie è altamente correlata e probabilmente vi è una componente deterministica dominante nella dinamica. Se l'ACF decresce rapidamente a zero, la serie è completamente stocastica o caotica. Quando $C_L(\tau)$ si annulla per la prima volta, abbiamo una buona indicazione nella scelta del time lag T . Per esempio, per la componente x dalla serie di Lorenz abbiamo un valore nullo per $n = 105$, come si vede dal grafico 2.4, che corrisponde, per un time step di 0.01, a 1.05 secondi. Tuttavia l'ACF rende conto solo delle dipendenze lineari e quindi, analizzando una

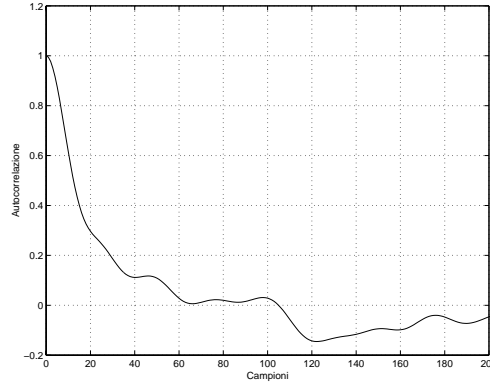


Figura 2.4: Autocorrelazione di una componente del sistema di Lorenz

serie temporale generata da un sistema fortemente non lineare, l'informazione che perdiamo utilizzando l'ACF potrebbe essere molto rilevante.

Una quantità chiamata *Mutua Informazione Media* (AMI) è una misura probabilistica che generalizza a domini non lineari la funzione di autocorrelazione lineare. Immaginiamo di avere due sistemi, A e B , che possono avere come valori a_i e b_k . La quantità di informazione, misurata in bits, che una misura di a_i porta ad una misura successiva di b_k è data dalla mutua informazione

$$I_{AB}(a_i, b_k) = \log_2 \left(\frac{P_{AB}(a_i, b_k)}{P_A(a_i)P_B(b_k)} \right) \quad (2.29)$$

dove la probabilità di osservare a_i da un set di valori di A è $P_A(a_i)$ e la probabilità di osservare b_k da un set di valori di B è $P_B(b_k)$. La probabilità congiunta di osservare a_i e b_k è $P_{AB}(a_i, b_k)$. La mutua informazione media tra una misura di un qualsiasi valore a_i di A e qualsiasi valore b_k di B è la media su tutte le possibili misure di $I_{AB}(a_i, b_k)$

$$I_{AB}(T) = \sum_{a_i, b_k} P_{AB}(a_i, b_k) I_{AB}(a_i, b_k) \quad (2.30)$$

Notiamo subito che $I(T) \geq 0$ e che se le misure dei sistemi A e B sono completamente indipendenti, $I_{AB}(a_i, b_k) = 0$.

Per utilizzare questa definizione astratta nel contesto della osservazione di un sistema fisico, possiamo considerare un set di misure $s(n)$ come appartenenti al processo A e lo stesso set ritardato di un time lag, $s(n+T)$, come appartenenti al processo B . Allora l'informazione media che l'osservazione $s(n)$ porta all'osservazione $s(n+T)$ è

$$I(T) = \sum_{n=1}^N P(s(n), s(n+T)) \log_2 \left[\frac{P(s(n), s(n+T))}{P(s(n))P(s(n+T))} \right] \quad (2.31)$$

Calcolare $I(T)$ è molto semplice. Per ottenere $P(s(n))$ dobbiamo valutare l'istogramma formato dalle frequenze con le quali si realizzano i valori di s . Se la funzione di distribuzione è stazionaria, $P(s(n))$ è uguale a $P(s(n+T))$. Per calcolare la probabilità congiunta dobbiamo valutare le frequenze nel piano $s(n), s(n+T)$, e poi normalizzare la distribuzione. Questo può essere realizzato facilmente, ed è abbastanza rapido anche con un PC.

Per stabilire un criterio di scelta per il time lag dobbiamo fare alcune considerazioni. Se T è troppo piccolo, la misura $s(n)$ ed $s(n+T)$ sono talmente correlate che non abbiamo bisogno di misurarle insieme. Se T è molto grande, $I(T)$ si avvicina a zero, non c'è alcuna correlazione statistica fra $s(n)$ e $s(n+T)$ e dunque non otteniamo alcuna utilità. Fraser e Swinney suggeriscono di scegliere il primo minimo dell'AMI per una ottima scelta del time lag [6]. In questo modo le misure $s(n)$ e $s(n+\tau)$ sono in qualche modo indipendenti, ma non dal punto di vista statistico. Per esempio, per la componente $x(t)$ dell'attrattore di Lorenz analizzato prima, possiamo notare che la mutua informazione media (figura 2.5) ha il primo minimo per $n = 10$ campioni. Quindi, con time step di 0.01 secondi, il grafico ci suggerisce un ritardo di 0.1 secondi.

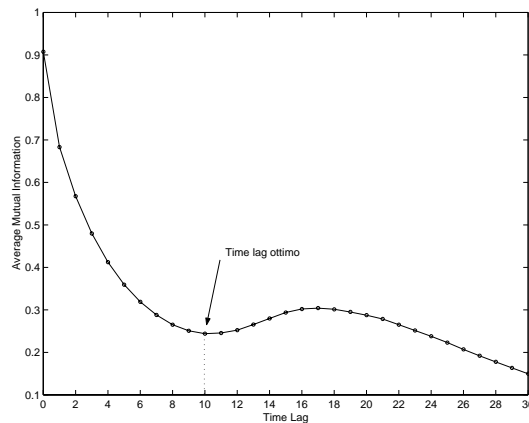


Figura 2.5: Mutua informazione media $I(T)$ in funzione del time lag per la serie di Lorenz

Può accadere, come avviene per la mappa di Hénon

$$\begin{aligned} x(n+1) &= 1 + y(n) - 1.4x(n)^2 \\ y(n+1) &= 0.3x(n) \end{aligned} \tag{2.32}$$

che l'AMI non ha alcun minimo. La ragione è molto semplice: l'intervallo di campi-

onamento τ_s è così lungo che i punti sull'orbita sono diventati scorrelati. Questo non significa che la mutua informazione media perde di significato, ma solo che bisogna utilizzare altri criteri di scelta: se sappiamo a priori che la serie è generata da una mappa, possiamo scegliere intuitivamente $T = 1$ o $T = 2$; oppure si può analizzare la funzione di autocorrelazione, scegliendo il punto in cui si annulla per la prima volta, per avere almeno una stima lineare del time lag; un altro criterio è scegliere τ cosicchè $I(T)/I(0) \approx \frac{1}{5}$. Questa è chiaramente una regola empirica suffragata dal fatto che in questo modo alcune effetti di decorrelazione sono dominanti, ma non troppo.

Scelta della dimensione di Embedding

Il teorema di embedding di Mañé e Takens si basa essenzialmente su risultati topologici. Lo scopo del teorema di ricostruzione è trovare uno spazio euclideo \mathbb{R}^d di dimensione minima tale che se un set di punti sono separati da una certa distanza nella dimensione d , accade per una proprietà tipica del sistema e non perchè è piccolo il numero di dimensioni dello spazio in cui il set viene osservato. Per esempio se l'attrattore è ad una dimensione, ed è osservato in uno spazio a due dimensioni, può ancora accadere che una linea intersechi se stessa in un punto isolato. In questo punto c'è ambiguità su quali punti sono vicini nella evoluzione temporale. Questa ambiguità è completamente risolta quando passiamo a $d = 3$, come suggerisce il teorema di Takens. Naturalmente passare a $d = 4$ significa rendere più chiara la distinzione. Non vengono aggiunte altre ambiguità passando a dimensione maggiore. Quando tutte le ambiguità sono risolte, allora lo spazio \mathbb{R}_E^d ha identificato l'attrattore.

Una lettura equivalente del teorema di embedding è il seguente. L'attrattore, di dimensione d_A , si sviluppa in una piccola parte dello spazio delle fasi. Noi possiamo sperare di effettuare una proiezione in un sottospazio nel quale l'attrattore può es-

sere fedelmente riprodotto. Se il sottospazio è di dimensione troppo piccola, allora si creano, come abbiamo notato, delle ambiguità dovute ai punti falsi vicini. In questo sottospazio l'orbita dell'attrattore è distorta e si genera una interpretazione fisica sbagliata. Invece la proiezione nel giusto sottospazio rende l'evoluzione dell'attrattore molto più regolare e chiara. Il teorema di embedding suggerisce una condizione sufficiente, ma non necessaria, per trovare la dimensione d_E .

La condizione sufficiente, tuttavia, non è sempre necessaria. Per esempio, per l'attrattore di Lorenz, $d_A = 2.06$. Il teorema di embedding suggerisce di scegliere $d_E > 2d_A$ e quindi $d_E = 5$. Ma le equazioni di Lorenz descrivono un sistema in \mathbb{R}^3 . Poichè noi stiamo discutendo di una versione ritardata di una variabile del sistema, e questa è una combinazione non lineare delle tre variabili fisiche, ci aspettiamo anche in questo caso di poter vedere l'attrattore gemello in uno spazio \mathbb{R}^3 di variabili ritardate. La differenza con il risultato suggerito dal teorema deriva dal fatto che si considerano solo le proprietà locali dello spazio delle fasi.

Chiaramente, dal punto di vista teorico nulla ci vieta di lavorare con spazi a dimensione maggiore rispetto a quelli sufficienti. Ma dal punto di vista pratico, vi sono due problemi principali:

1. quasi tutti gli algoritmi utilizzati per estrarre informazioni dai dati richiedono ricerche ed altre operazioni il cui costo computazionale incrementa esponenzialmente con d .
2. in presenza di rumore, la dimensione extra non contiene aspetti determinanti della dinamica, catturata già a dimensioni minori, ma solo contaminazioni sull'osservazione. Ed è quindi inutile indagare in questi sottospazi.

Ora è chiaro per quale motivo è importante studiare un metodo per identificare una necessaria dimensione di embedding. Due sono i metodi che verranno discussi:

Singular-Value-Decomposition (SVD) della matrice di covarianza dei campioni ed il metodo del False Nearest Neighbors (FNN).

Il Single-value analysis si basa sulla considerazione che un processo stocastico ha dimensione teorica di embedding infinita, mentre un processo deterministico ha dimensione di embedding finita, ed in genere bassa.

Se la nostra misura $y(n)$ è composta dal segnale generato dal sistema dinamico che vogliamo studiare più una contaminazione di un altro sistema, in assenza di informazione specifica è plausibile pensare che sia più difficile cogliere la dinamica a dimensione più alte. Chiamiamo con d_N la dimensione necessaria per rappresentare la dinamica. Se scegliamo $d_E > d_N$ allora, dal punto di vista euristico $d_E - d_N$ dimensioni sono popolate solo dalla contaminazione. Se immaginiamo che nello spazio a d_E dimensioni il segnale è composto dal segnale da noi cercato $s(n)$ più una contaminazione $c(n)$ cosicché $y(n) = s(n) + c(n)$, allora la matrice di covarianza di dimensione $d_E \times d_E$

$$COV = \frac{1}{N} \sum_{n=1}^N [y(n) - \bar{y}][y(n) - \bar{y}]^T \quad (2.33)$$

con

$$\bar{y} = \frac{1}{N} \sum_{n=1}^N y(n) \quad (2.34)$$

avrà, sempre dal punto di vista euristico, d_N autovalori che rappresentano le variazioni del segnale reale dalla sua media e $d_E - d_N$ autovalori che rappresentano il rumore. Se il rumore è rappresentato in dimensioni più alte, possiamo immaginare che le riempie in un modo abbastanza uniforme e si possono identificare i $d_E - d_N$ autovalori simili osservando quando lo spettro degli autovalori diventa piatto. In più la single value analysis permette, proiettando la serie nel sottospazio a dimensione d_E , di effettuare una prima parte di denoising.

Questa tecnica offre i vantaggi di essere facilmente implementabile, ma è di difficile interpretazione. Inoltre, a causa della struttura lineare dell'algoritmo, la

SVD non distingue processi che hanno lo stesso spettro di Fourier. Tuttavia è una tecnica che, in aggiunta ad altre più efficaci, può dare una indicazione in più per la ricerca della dimensione di embedding.

Il False nearest neighbors prende spunto proprio dalla domanda principale del teorema di embedding: quando sono eliminate le false vicinanze dell'orbita con se stessa, dovute ad una proiezione in un sottospazio di dimensione troppo piccola? Esaminando la domanda prima in una dimensione, poi in due, e così via, si può stabilire, solo con considerazioni geometriche, il valore della dimensione di embedding necessaria.

Nella dimensione d ogni vettore

$$\mathbf{y}(k) = [s(k), s(k+T), \dots, s(k+(d-1)T)] \quad (2.35)$$

avrà un altro vettore \mathbf{y}^{NN} più vicino di altri, secondo una certa distanza. Utilizziamo la distanza Euclidea in d dimensioni che per i vettori $\mathbf{y}(k)$ e $\mathbf{y}^{NN}(k)$ si scrive

$$\begin{aligned} R_d^2(k) &= [s(k) - s^{NN}(k)]^2 + [s(k+T) - s^{NN}(k+T)]^2 + \dots \\ &\dots + [s(k+(d-1)T) - s^{NN}(k+(d-1)T)]^2 \\ &= \sum_{m=1}^d [s(k+(m-1)T) - s^{NN}(k+(m-1)T)]^2 \end{aligned} \quad (2.36)$$

La distanza è piccola per piccoli data set e per N dati è dell'ordine di $1/N^{1/d}$.

Nella dimensione $d+1$ questa distanza cambia e, a causa della $(d+1)^{st}$ coordinata $s(k+dT)$ e $s^{NN}(k+dT)$, diventa

$$R_{d+1}^2(k) = R_d^2(k) + [s(k+dT) - s^{NN}(k+dT)]^2 \quad (2.37)$$

Se $R_{d+1}(k)$ è grande, si può presumere che la vicinanza dei due punti è dovuta ad una proiezione dell'attrattore in un sottospazio troppo piccolo: passando da un sottospazio a dimensione d ad uno a dimensione $d+1$ abbiamo ripristinato la corretta distanza fra i due punti. Se

$$\frac{|s(k+dT) - s^{NN}(k+dT)|}{R_d(k)} > R_T \quad (2.38)$$

dove R_T è una soglia empirica, allora viene decretato un false neighbor. In pratica per valori di R_T nel range $10 - 50$ il numero dei false neighbors è costante. Quindi in questo range si ha l'indipendenza dal parametro R_T . Quando il numero di false nearest neighbors va a zero, allora abbiamo la giusta dimensione di embedding.

Applicando questo criterio al sistema di Lorenz si ottiene giustamente $d_E = 3$ (figura 2.6) anche se la condizione sufficiente suggeriva $d_E = 5$.

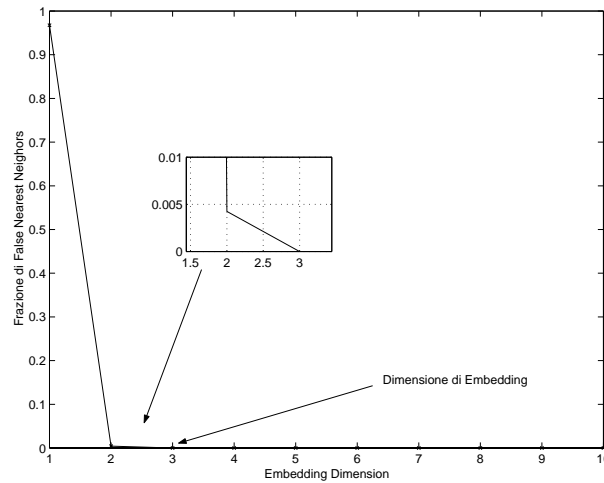


Figura 2.6: Percentuale di FNN per l'attrattore di Lorenz

Puntualizziamo ora un altro aspetto. Il rumore è comunemente visto come completamente non predicibile e casuale. Questo è tipico di un processo con spettro piatto di Fourier e con funzione di autocorrelazione che decresce rapidamente a zero. Poichè queste caratteristiche nello spettro e nella funzione di autocorrelazione sono tipiche anche dei processi caotici a bassa dimensionalità, con l'analisi standard non è possibile distinguere un sistema caotico da un processo stocastico. Abbiamo tuttavia visto che la tecnica del FNN permette di distinguere in quanto il rumore appare solo come caos ad alta dimensionalità. Quando si usa il test del false-nearest-neighbors su una serie generata da un sistema dinamico che è stata corrotta dal rumore, o da segnali di un altro sistema dinamico (con dimensionalità maggiore o minore), la

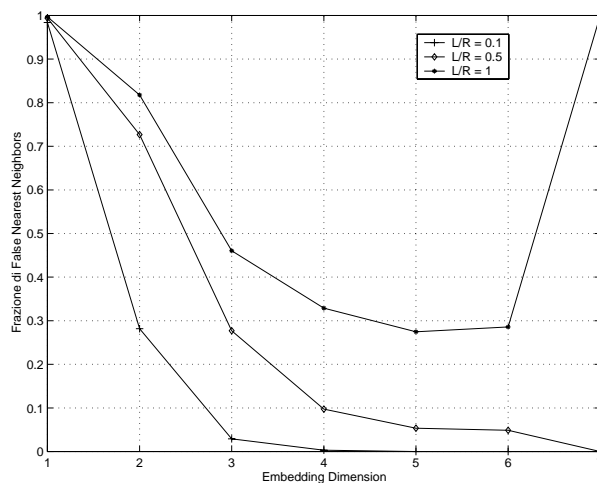


Figura 2.7: Percentuale di FNN per l'attrattore di Lorenz per diversi livelli di rumore

tecnica del FNN risulta molto robusta. Per esempio, in figura 2.7, possiamo vedere il grafico della percentuale dei FNN in funzione della dimensione di embedding a diversi livelli di rumore.

Il minimo del grafico del FNN rappresenta la dimensione di embedding del sistema. Successivamente si nota una risalita dovuta alla presenza di rumore nella serie, che come detto, si manifesta soprattutto ad alta dimensionalità.

Bibliografia relativa al capitolo II

- [1] Henry D.I. Abarnabel, *Analysis of Observed Chaotic Data*, Springer, 1996
- [2] Henry D.I. Abarnabel, *The analysis of observed chaotic Data in physical system*, Review of Modern Physics, Vol.64 No. 4, Ottobre 1993
- [3] R. Hegger, H. Kantz, T. Schreiber, *Practical implementation of nonlinear time series methods: the TISEAN package*
- [4] T. Schreiber, *Interdisciplinary application of non linear time series methods*
- [5] Cover-Thomas, *Elements of Information Theory*, J.Wiley, 1991
- [6] Oppenheim-Schafer, *Discrete-time Signal Processing*, Prentice-Hall, 1989
- [7] S.M. Ross, *Introduction to probability Models*, Academic Press, 1989
- [8] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGRAW-HILL Book Company
- [9] H.Kantz ed altri, *Non linear noise reduction: A case study on experimental data*, Physical Review E, vol. 48, N. 2, Agosto 1993
- [10] B.F. Schutz, *Introduction to the analysis of low-frequency gravitational wave data*, gr-qc/9710080
- [11] B. J. Owen, B.S. Sathyaprakash, *Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement*, gr-qc/9808076
- [12] S. Mukherjee, L.S. Finn, *Removing instrumental artifacts: suspension violin modes*, gr-qc/9911098

Capitolo 3

Reti Neurali

In questo capitolo vengono introdotti alcuni concetti di base sulle reti neurali utili per meglio comprendere il mio lavoro di tesi in questo ambito. Pur non trattando i principi degli approcci alternativi, sono introdotti ove necessario i risultati ottenuti con altre tecniche.

A partire dalla fine degli anni '80 c'è stata un'enorme crescita degli studi effettuati nel campo delle reti neurali, non solo per quanto riguarda lo sviluppo di nuovi modelli e nuovi algoritmi, ma anche per quanto riguarda le applicazioni al mondo reale. È attualmente riconosciuto che le reti neurali offrono un insieme potente di strumenti per poter risolvere problemi di pattern recognition, elaborazione di dati e controllo non lineare, da utilizzare parallelamente o in sostituzione ai metodi convenzionali.

Dei vari tipi di reti neurali oggi esistenti, le più applicate sono rispettivamente i perceptron a strati (*MLP Multilayer Perceptrons*) e le reti a funzioni di base radiali (*RBF Radial Basis Function Net*); questi modelli fanno parte della classe generalmente nota con il nome di reti feedforward in quanto l'informazione procede in una direzione, senza effetti di feedback. Una rete neurale feedforward può essere vista come una funzione matematica non lineare che trasforma un insieme di variabili di

ingresso in uno di uscita. La forma precisa della trasformazione è indicata dall'insieme di parametri chiamati pesi (o connessioni o sinapsi), il cui valore può essere determinato sulla base di un insieme di esempi della funzione che si vuol ricostruire. Il processo di determinazione del valore dei pesi è spesso chiamato learning o training, ed è la parte computazionalmente più onerosa. Al contrario, una volta fissati i pesi i nuovi dati possono essere processati in modo assai rapido.

Il principale svantaggio delle reti neurali emerge dalla necessità di fornire alla rete un insieme di dati di esempio sufficientemente rappresentativo per poter portare a termine l'apprendimento, altrimenti sia l'interpolazione che l'estrapolazione a nuovi punti dello spazio degli ingressi può portare ad una scarsa efficienza computazionale.

In altre parole le reti neurali dovrebbero essere considerate gli strumenti ideali per quei problemi dove:

- esiste un ampio insieme di apprendimento
- è difficile o impossibile fornire con tecniche standard una soluzione adeguata
- i nuovi dati devono essere elaborati ad alta velocità
- il metodo di elaborazione dei dati deve sopportare un modesto livello di rumore dei dati di ingresso

Dai quattro punti precedenti emerge la ragione per cui le reti neurali stanno avendo un'ampia applicazione nell'ambito dell'estrazione di informazione da segnali di cui non si conosce perfettamente l'origine.

3.1 Fondamenti teorici - il neurone

Il neurone è il mattone con cui si costruisce qualsiasi rete neurale. Il primo modello matematico fu introdotto da McCulloch e Pittis, e può essere considerato come

una funzione non lineare che trasforma un insieme di variabili di ingresso x_i , ($i = 1, \dots, d$), in una variabile di output z (vedi figura 3.1).

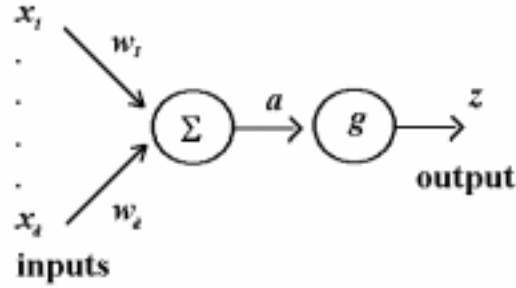


Figura 3.1: Modello di neurone di Mc Culloch e Pittis

I valori x_i dei vari input, sono prima moltiplicati per un parametro w_i e poi sommati. Il risultato è l'input totale del neurone:

$$a = \sum_{i=1}^d w_i x_i + w_0 \quad (3.1)$$

dove il termine w_0 è chiamato soglia interna del neurone. Formalmente tale soglia può essere considerata come input extra con valore costante pari ad uno. In questo modo l'equazione 3.1 può essere riscritta come

$$a = \sum_{i=0}^d w_i x_i \quad (3.2)$$

L'uscita z del neurone è poi trasformato da una funzione non lineare g (detta di *attivazione*) che agisce su a :

$$z = g(a) = g\left(\sum_{i=0}^d w_i x_i\right) \quad (3.3)$$

Questo semplice elemento è il mattone attraverso cui vengono costruite quasi tutte le reti neurali. Dal collegamento di tali semplici unità di calcolo è possibile

generare classi di funzioni non lineari particolarmente generali e complesse, che sono applicabili alla soluzione di problemi non risolvibili con metodi classici. Il processo di apprendimento di tali reti è quindi quello di adattare, per mezzo di un opportuno algoritmo, i pesi che connettono i vari neuroni in modo da fornire una corretta risposta ad un dato input.

Il problema della determinazione del valore dei pesi è effettuato spesso tramite un supervisore esterno, perciò queste reti sono dette *supervised*. In generale tale supervisore non è altro che una funzione dell'errore commesso dalla rete nel valutare un certo pattern di input x^q , cioè una funzione della differenza fra l'output della rete e l'output desiderato. Esistono varie funzioni di errore in letteratura, ma la più comune (ed in seguito da me utilizzata) è lo scarto quadratico medio (*RMS*). Se indichiamo la risposta della rete con $y(x^q; \mathbf{w})$, la forma della funzione di errore risulta

$$J = \frac{1}{2} \sum_{q=1}^n \{y(x^q; \mathbf{w}) - t^q\}^2 \quad (3.4)$$

in questo caso la sommatoria si estende a tutti gli n dati di ingresso, mentre t^q è il valore target o desiderato. Il processo di apprendimento di una rete neurale procede quindi minimizzando la funzione di errore scelta (per esempio quella definita dall'equazione 3.4) rispetto ai parametri \mathbf{w} che sono le connessioni dei vari neuroni. Nel caso generale la rete neurale può avere più di un neurone di uscita per cui l'equazione. 3.4 diviene

$$J = \frac{1}{2} \sum_{q=1}^n \sum_{k=1}^c \{y_k(x^q; \mathbf{w}) - t_k^q\}^2 \quad (3.5)$$

in cui l'indice k indica il neurone di uscita e t_k^q il corrispondente valore desiderato.

3.2 Architetture principali: reti statiche

Nel paragrafo precedente è stato introdotto il concetto di singola unità computazionale o neurone. Se si considera un insieme m di tali unità aventi l'ingresso in comune, si arriva ad avere una rete neurale avente un singolo strato di pesi adattabili come è illustrato in figura 3.2.

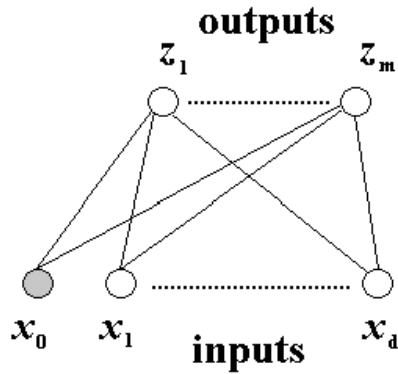


Figura 3.2: Modello di rete feedforward ad un singolo strato di connessioni

Le variabili di uscita indicate con z_j sono date da

$$z_j = g(a_j) = g\left(\sum_{i=0}^d w_{ji}x_i\right) \quad (3.6)$$

dove w_{ji} è il peso che connette l'input i e l'output j , e $g(\cdot)$ è una funzione di attivazione. Si può notare che la figura 3.2 è quindi la rappresentazione grafica dell'equazione 3.6, in cui ciascun cerchio nello strato basso rappresenta x_i , ciascun cerchio nello strato in alto rappresenta l'attivazione del neurone z_j e le linee che connettono i cerchi sono i pesi w_{ji} , mentre l'input extra x_0 (o soglia) è rappresentato in figura con un cerchio pieno.

3.2.1 Reti MLP

Queste reti a singolo strato dette *perceptrons* furono estensivamente studiate negli anni '60. Queste reti hanno una capacità di elaborazione limitata¹ che può essere superata generando reti multistrato (MPL) ottenute sovrapponendo più strati di pesi (come mostrato in figura 3.3).

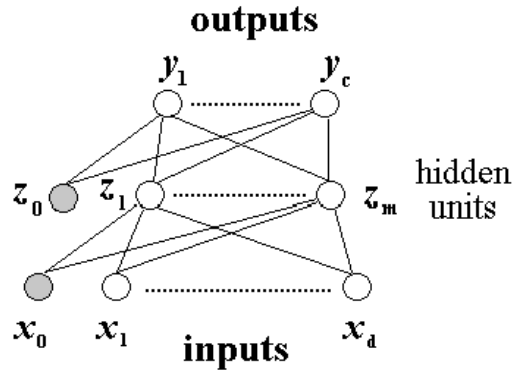


Figura 3.3: Modello di rete MLP

Questo tipo di rete non fu studiata a fondo negli anni '60 perché mancava un algoritmo di apprendimento generale che permettesse una variazione dei pesi rispetto all'output esterno. Lo strato di neuroni intermedio è di solito chiamato *hidden*, perché il loro livello di attivazione non è direttamente accessibile dall'esterno della rete. L'attivazione di queste unità nascoste è ancora dato dall'equazione 3.6, mentre il livello di attivazione delle uscite è ottenuto attraverso una seconda trasformazione

$$y_k = \tilde{g}(\tilde{a}_k) = \tilde{g}\left(\sum_{j=0}^m \tilde{w}_{kj} z_j\right) \quad (3.7)$$

in cui \tilde{w}_{kj} indica il peso che riguarda la nuova connessione, dalla unità nascosta j a

¹Il problema riguarda la realizzazione o meno delle condizioni di applicabilità del teorema di Kolmogorov. Una rete con un singolo strato può risolvere solo i problemi detti linearmente separabili perché può approssimare solo una funzione lineare nello spazio delle features.

quella di uscita k , mentre con $\tilde{g}(\cdot)$ si è indicata la nuova funzione di trasferimento. z_j indica $g(a_j)$ e anche in questo caso si è inserita la soglia nello strato dei pesi come neurone nascosto sempre attivato $z_0 = 1$.

Se si combina l'equazione 3.7 con la equazione 3.6 si rende esplicita l'espressione completa della trasformazione non lineare operata dalla rete neurale dallo spazio degli ingressi a quello delle uscite

$$y_k = \tilde{g} \left(\sum_{j=0}^m \tilde{w}_{kj} g \left(\sum_{i=0}^d w_{ji} x_i \right) \right) \quad (3.8)$$

Da tale trasformazione si può notare che se la funzione di attivazione viene scelta lineare il tutto si riduce al semplice prodotto di matrici. Perciò per poter rendere più generale la funzione di uscita vengono prese le funzioni di attivazione non lineari e le due più utilizzate sono

$$\begin{aligned} g(a) &= \tanh(\beta a) \\ g(a) &= \frac{1}{1 + \exp(-\beta a)} \end{aligned}$$

Tali funzioni hanno anche l'interessante proprietà che la loro derivata può essere espressa in termini della funzione stessa. Inoltre, a differenza del perceptrone, è stato dimostrato che le reti *MLP* hanno elevate capacità computazionali e possono approssimare tutte le funzioni regolari. Il parametro β determina la linearità della funzione di attivazione.

3.2.2 Reti RBF

Una delle limitazioni delle *MPL* riguarda il fatto che il processo di apprendimento può essere computazionalmente molto intenso. Poiché in pratica per poter scegliere la rete migliore bisogna ripetere tale processo un certo numero di volte, la fase di apprendimento può diventare talvolta un problema. Una seconda difficoltà può emergere dalla rappresentazione interna degli *MLP*, che è di solito impossibile da

interpretare e qualche volta ciò può essere richiesto per qualche applicazione particolare. In questo paragrafo sono presentate le reti neurali a funzioni radiali di base (*RBF*), che tentano di superare i due problemi sopra esposti.

Le reti *RBF* sono basate sulla semplice ed intuitiva idea che una funzione arbitraria $y(x)$ può essere approssimata da una combinazione lineare di un insieme di funzioni di base localizzate $\phi_j(x)$, come è schematicamente indicato in figura 3.4.

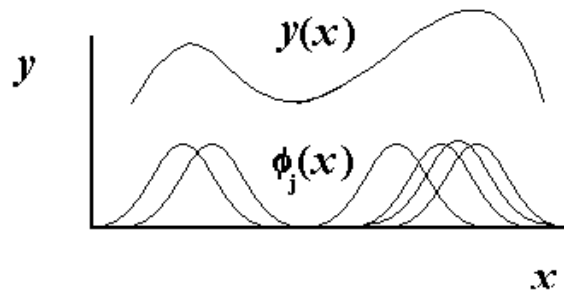


Figura 3.4: Esempio di approssimazione con RBF

Le *RBF* hanno origine dalle tecniche usate per l'esatta interpolazione tra punti di dati ad alta dimensionalità. Ciò è ottenuto rappresentando le uscite di una rete come combinazione lineare di funzioni di base, una per ciascun punto del training set nella forma:

$$y_k = \sum_{q=1}^n \tilde{w}_{kq} \phi_q(x) \quad (3.9)$$

dove $\phi_q(x)$ è una funzione radiale simmetrica centrata del dato *q-esimo*. L'interpolazione esatta richiede che i valori degli y_k riproducano il valore desiderato esattamente. Ciò conduce alla risoluzione di un sistema di n equazioni ad n incognite.

Nell'applicazione corrente delle reti neurali, però si è di solito interessati a trovare una rappresentazione *smooth* dell'andamento, piuttosto che l'esatta interpolazione dei dati; in particolare se i dati del training sono rumorosi l'esatta interpolazione

porta ad una pessima capacità di generalizzazione. Una interpolazione più regolare può essere fatta minimizzando la somma dei quadrati degli errori (come per *MLP*), utilizzando meno funzioni di base di quanti siano i punti dell'insieme di apprendimento. Questo conduce ad un problema sovradeterminato rappresentato da una rete neurale con output:

$$y_k = \sum_{j=1}^m \tilde{w}_{kj} \phi_j(x) \quad (3.10)$$

con $m < n$. In questo caso $\phi_j(x)$ rappresenta l'attivazione del nodo nascosto j quando alla rete è presentato in ingresso il vettore di input x . Nella maggior parte delle applicazioni le funzioni di base radiali sono scelte gaussiane del tipo

$$\phi_j = \left\{ -\frac{|x - \mu_j|^2}{2\sigma_j} \right\} \quad (3.11)$$

In cui μ_j rappresenta il centro della j -esima funzione di base e σ_j è la relativa ampiezza. In figura 3.5 è rappresentata una rete *RBF* in cui ciascuno dei nodi nascosti rappresenta una funzione di base, il cui fascio di pesi in ingresso è sostituito dal vettore μ_j e la soglia interna dal parametro $\sigma = \sigma_j$, che può ritenersi, in prima approssimazione, uguale per tutte le RBF. Il secondo livello di pesi è analogo a quello di un *MPL* che ha le funzioni di attivazione di uscita lineari.

Il procedimento che rende l'apprendimento di questa rete molto veloce, è quello di dividere il problema in due parti. Nella prima fase si scelgono (e si fissano) i valori ottimali per i centri delle funzioni di base e le loro ampiezze con qualche algoritmo di clustering. Nella seconda fase, determinati i parametri delle gaussiane, si cercano i coefficienti (o pesi del secondo strato) che minimizzano l'errore dell'insieme di apprendimento. Esistono vari algoritmi per la ricerca della soluzione migliore delle due fasi da molto semplici ad estremamente complessi e per una trattazione completa ci si può riferire a [1].

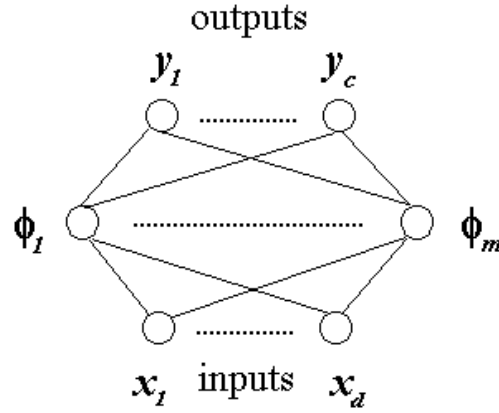


Figura 3.5: Rete RBF

3.3 Architetture principali: reti dinamiche

Le reti neurali feedforward, sia nel modello MPL che in quello RBF, possono apprendere solo una trasformazione immediata tra input ed output, ma non sono in grado di estrarre informazioni temporali. Tuttavia in molte applicazioni l'input possiede una struttura temporale la cui identificazione è essenziale per ottenere dalla rete una risposta corretta [2], [3], [5].

Usiamo il seguente formalismo per una MLP con un neurone di output: il modello può essere espresso dalle equazioni

$$y = f(\mathbf{c}^T \mathbf{z} + c_0) \quad (3.12)$$

$$\mathbf{z} = F_n(B\mathbf{u} + \mathbf{b}_0) \quad (3.13)$$

dove \mathbf{z} è un vettore n dimensionale che indica l'uscita dei neuroni hidden, y è uno scalare che indica il valore del neurone di output, \mathbf{u} è un vettore m dimensionale che indica gli input della rete, c_0 e \mathbf{b}_0 indicano rispettivamente il valore di soglia del neurone di output e dei neuroni hidden, B è una matrice $n \times m$ che denota i valori delle connessioni fra lo strato input ed hidden, $f(\cdot)$ è la funzione di attivazione non lineare,

infine $F_n(\cdot)$ è un vettore n dimensionale della forma $F_n(\cdot) = [f(\cdot), f(\cdot), \dots, f(\cdot)]^T$. Il numero totale dei parametri è $n \times m + n + n + 1 = n(m + 2) + 1$. Il formalismo può essere facilmente esteso anche a reti con output multidimensionale. Vediamo ora alcuni modelli di reti neurali ricorrenti proposti in diversi ambiti di ricerca.

3.3.1 Time Delay Neural Networks

Un modo semplice per riconoscere serie temporali consiste nel trasformare la dimensione temporale in dimensione spaziale, ovvero nello scegliere il numero di unità di input in modo da coprire l'intervallo temporale in cui il segnale viene campionato a istanti discreti, come se il segnale scorresse lungo la finestra di input (figura 3.6). I

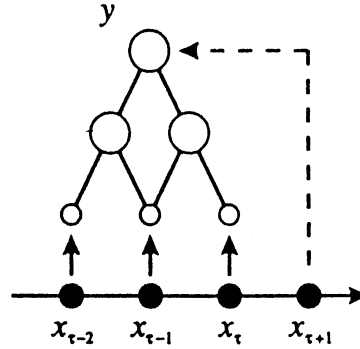


Figura 3.6: Schema di rete TDNN

valori di attivazione delle unità interne vengono infine trasmessi, pesati e sommati, alle unità di output.

Questo tipo di architettura richiede la conoscenza del numero adeguato di unità di input, cioè di un numero ottimale di ritardi, che può essere associato alla dimensione di embedding della serie temporale. Inoltre le reti TDNN mantengono le caratteristiche di robustezza ed efficienza tipiche delle reti feedforward.

3.3.2 Reti ricorrenti in forma canonica

Un altro metodo per codificare informazioni temporali è cambiare la struttura di una rete MLP con il seguente modello:

1. un sotto modello feedforward, come una rete MLP standard.
2. un sotto modello feedback, che include solo d delay. Se l'output della parte feedforward è $y(t)$, e l'ingresso è $u(t)$, allora la parte feedback può rappresentarsi nel seguente modo:

$$\begin{aligned}
 v_0 &= y(t) \\
 v_1(t) &= v_0(t-1) \\
 v_2(t) &= v_1(t-1) \\
 &\vdots = \vdots \\
 v_d(t) &= v_{d-1}(t-1)
 \end{aligned} \tag{3.14}$$

Questi d segnali di feedback, $v_i(t), i = 1, 2, \dots, d$ possono essere usati come input aggiuntivi della parte feedforward, come si vede in figura 3.7.

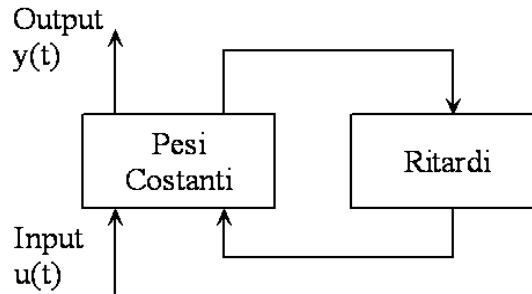


Figura 3.7: Forma canonica di una rete ricorrente

In questo modo l'architettura è composta da due parti:

1. una parte feedforward in cui l'attivazione dei neuroni hidden viene modificata nel seguente modo:

$$\mathbf{z}(t) = F_n \left([B_1 \quad B_2] \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{v}(t) \end{bmatrix} + \mathbf{b}_0 \right) \quad (3.15)$$

dove $\mathbf{v}(t)$ è un vettore d -dimensionale le cui componenti sono date dalla 3.14. B_1 e B_2 sono rispettivamente matrici $n \times m$ e $n \times d$.

2. una parte feedback descritta dal vettore $\mathbf{v}(t)$.

Da notare che l'architettura ha una esplicita memoria di d ritardi temporali, ma dipendente dai valori delle matrici dei pesi e dei valori delle soglie di attivazione. Per questo motivo, implicitamente, l'influenza dei valori passati è sull'output corrente può essere più persistente rispetto ai d ritardi. Questa struttura prende anche il nome di rete di Jordan.

3.3.3 Modelli NARMAX

Un modello molto popolare nell'analisi lineari di serie temporali è il modello autoregressivo a media mobile (*ARMA*).

$$y(t) = \sum_{i=1}^{n_a} a_i y(t-i) + \sum_{i=0}^{n_b} b_i x(t-i) + \varepsilon(t) \quad (3.16)$$

dove $\varepsilon(t)$ indica l'errore del modello a produrre l'output $y(t)$. n_a ed n_b sono le dimensioni del modello e $a_i, i = 1, \dots, n_a$ e $b_i, i = 1, \dots, n_b$ sono i parametri da stimare. È possibile generalizzare questo modello includendo la non linearità ed input esterni in modo da avere la struttura più generale possibile. Questi modelli sono detti *NARMAX*².

$$y(t) = \mathcal{F} \left(\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{i=0}^{n_b} b_i x(t-i) + \sum_{i=0}^{n_d} d_i \varepsilon(t-i) \right) \quad (3.17)$$

²Non linear **A**uto **R**egressive **M**oving **A**verage with e**X**ogeneous input

Queste equazioni sono state scritte per una variabile di input ed una di output, ma possono facilmente essere generalizzate a più variabili. Si può dimostrare che questo modello è equivalente, dal punto di vista computazionale, alla macchina di Turing, e ha proprietà di approssimazione universale.

3.3.4 Modello FCHLRNN

Un modello di rete ricorrente più generale è quella in cui lo strato di neuroni hidden è totalmente connesso (FCHLRNN: Full Connected Hidden Layer Recurrent Neural Network). In questo caso il modello può essere rappresentato dall'equazione

$$\mathbf{z}(t) = F_n(A\mathbf{z}(t-1) + B\mathbf{u}(t) + \mathbf{b}_0) \quad (3.18)$$

dove A è una matrice $n \times n$. Dalla struttura della matrice A dipende essenzialmente l'architettura. Per esempio lo strato hidden può essere connesso ad anello e la matrice A ha struttura

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 & \dots & 0 & a_{1n} \\ a_{21} & 0 & a_{23} & 0 & 0 & \dots & 0 & 0 \\ 0 & a_{32} & 0 & a_{34} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & 0 & 0 & 0 & 0 & \dots & a_{n,n-1} & 0 \end{bmatrix} \quad (3.19)$$

Il valore principale di questa architettura è che soddisfa le condizione di approssimatore universale.

Possiamo incorporare nella struttura anche d ritardi, generalizzando l'equazione 3.18, che diventa

$$\mathbf{z}(t) = F_n \left(\sum_{i=0}^d A_i \mathbf{z}(t-i) + B\mathbf{u}(t) + \mathbf{b}_0 \right) \quad (3.20)$$

dove A_i sono d matrici $n \times n$. Possiamo infine incorporare anche una connessione diretta fra l'input e l'output. In questo caso l'equazione per il neurone di

processo di apprendimento consiste nel cercare il minimo (idealmente globale) di tale funzione rispetto alla variazione dei pesi della rete. In teoria si può partire valutando prima il gradiente della funzione di errore e poi calcolando l'Hessiano di tale funzione per avere una informazione topologica corretta. Tale procedimento è però estremamente oneroso dal punto di vista computazionale e di fatto applicabile soltanto a reti molto piccole.

Nel 1986 è stato introdotto un algoritmo, per calcolare le derivate della funzione di errore, chiamato *Backpropagation* che, pur non essendo molto efficiente nella ricerca del minimo globale, è applicabile a reti di dimensione arbitraria. È con l'introduzione di tale algoritmo che di fatto le reti neurali sono passate da un campo di pura indagine teorica a quello più applicativo.

3.4.1 Metodi basati sul gradiente

In questa sezione deriveremo informazioni legate al gradiente. Useremo una tecnica legata all'approccio variazionale usato nella teoria dell'ottimizzazione [1], [4].

Supponiamo di avere un training data set $\{d(t), \mathbf{u}(t)\}$, con $t = 1, 2, \dots, N$, essendo $d(t)$ l'output desiderato al tempo t . Il nostro scopo è la realizzazione di un algoritmo di apprendimento in grado di stimare i parametri non noti in una architettura RNN, usando il training data set, una volta definita la struttura della rete (tipo di architettura, numero di neuroni hidden, tipo di sinapsi dinamiche e ordine dei filtri corrispondenti). Come criterio di errore utilizziamo l'errore quadratico medio, come avevamo già precisato

$$J(\theta) = \frac{1}{2} \sum_{i=1}^N (d(t) - y(t))^2 \quad (3.22)$$

dove θ è un vettore di dimensione n_θ le cui componenti sono i pesi della rete. Una stima del valore di θ può essere ottenuta minimizzando la funzione di errore.

Poichè dal punto di vista analitico questa operazione è molto onerosa, possiamo

trovare una soluzione numerica usando un semplice algoritmo per il gradiente del primo ordine. In prima approssimazione il gradiente può scriversi

$$J(\theta + \delta\theta) \approx J(\theta) + (\delta\theta)^T J'(\theta) \quad (3.23)$$

Se scegliamo $\delta\theta = -\eta J'(\theta)$, con $\eta \geq 0$, otteniamo

$$J(\theta + \delta\theta) \approx J(\theta) - \eta (J'(\theta))^T J'(\theta) \quad (3.24)$$

Allora i parametri θ possono essere aggiornati con la regola

$$\hat{\theta}_k = \hat{\theta}_{k-1} - \eta_k \left. \frac{\partial J}{\partial \theta} \right|_{\theta=\hat{\theta}_{k-1}} \quad (3.25)$$

dove $\hat{\theta}_k$ indica la stima dei parametri alla k-esima iterazione. η_k è il parametro di apprendimento, detto learning rate, e può essere variabile nel tempo. Appare ovvio che il primo obiettivo è calcolare il gradiente di $J(\theta)$. L'idea di base, come si vede in figura 3.9, è di aggiornare i parametri nella direzione negativa del gradiente per trovare il minimo, possibilmente globale.

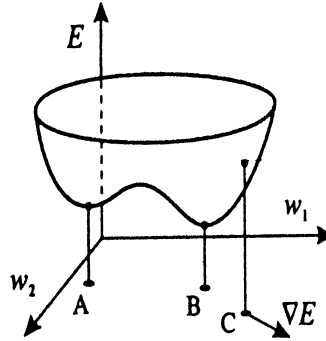


Figura 3.9: Superficie di errore con direzione indicata del gradiente

3.4.2 Algoritmo di apprendimento per FCHLRNN

Valuteremo prima il gradiente di J rispetto ai parametri \mathbf{c} , a_{ij} che sono gli elementi della matrice A , e poi b_{ij} che sono elementi della matrice B .

La derivata della funzione di costo rispetto a \mathbf{c} è

$$\begin{aligned}\frac{\partial J}{\partial c_i} &= -\sum_{t=1}^N (d(t) - y(t)) \frac{\partial y(t)}{\partial c_i} \\ &= -\sum_{t=1}^N (d(t) - y(t)) z_i(t)\end{aligned}\quad (3.26)$$

dove $z_i(t)$ è l'elemento i -esimo del vettore $\mathbf{z}(t)$. In modo compatto possiamo scrivere

$$\frac{\partial J}{\partial \mathbf{c}} = -\sum_{t=1}^N (d(t) - y(t)) \mathbf{z}(t) \quad (3.27)$$

La derivata rispetto a c_0 si effettua allo stesso modo ed otteniamo

$$\frac{\partial J}{\partial c_0} = -\sum_{t=1}^N (d(t) - y(t)) \quad (3.28)$$

Similmente per la derivata rispetto a a_{ij} possiamo scrivere:

$$\begin{aligned}\frac{\partial J}{\partial a_{ij}} &= -\sum_{t=1}^N (d(t) - y(t)) \frac{\partial y(t)}{\partial a_{ij}} \\ &= -\sum_{t=1}^N (d(t) - y(t)) \sum_{k=1}^n c_k \frac{\partial z_k}{\partial a_{ij}}\end{aligned}\quad (3.29)$$

Possiamo calcolare esplicitamente la derivata al secondo termine ed ottenere

$$\frac{\partial z_k}{\partial a_{ij}} = f'(x_k) \left[a_{k1} \frac{\partial z_1}{\partial a_{ij}} + a_{k2} \frac{\partial z_2}{\partial a_{ij}} + \dots + a_{kn} \frac{\partial z_n}{\partial a_{ij}} \right] \quad (3.30)$$

dove x_k è il k -esimo elemento del vettore definito da

$$\mathbf{x}(t) = A\mathbf{z}(t-1) + B\mathbf{u}(t) \quad (3.31)$$

Mettendo insieme i termini possiamo scrivere in modo più compatto

$$\frac{\partial \mathbf{z}(t)}{\partial a_{ij}} = \Lambda(\mathbf{x}(t)) \left[A \frac{\partial \mathbf{z}(t-1)}{\partial a_{ij}} + Q_{ij} \mathbf{z}(t-1) \right] \quad (3.32)$$

dove Q_{ij} è una matrice $n \times n$ con tutti gli elementi nulli tranne un 1 nella posizione ij . La matrice $\Lambda(\mathbf{x}(t))$, di dimensione $n \times n$ è definita come

$$\Lambda(\mathbf{x}(t)) = \begin{bmatrix} f'(x_1(t)) & 0 & 0 & 0 & \dots & 0 \\ 0 & f'(x_2(t)) & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & f'(x_n(t)) \end{bmatrix} \quad (3.33)$$

Definendo $\psi_{ij}(t) \triangleq \frac{\partial \mathbf{z}(t)}{\partial a_{ij}}$ possiamo riscrivere l'equazione 3.32 come

$$\psi_{ij}(t) = \Lambda(\mathbf{x}(t))[A\psi_{ij}(t-1) + Q_{ij}\mathbf{z}(t-1)] \quad (3.34)$$

con condizione iniziale $\psi_{ij}(0) = 0$, in questo caso abbiamo

$$\frac{\partial J}{\partial a_{ij}} = - \sum_{t=1}^N (d(t) - y(t)) \mathbf{c}^T \psi_{ij}(t) \quad (3.35)$$

per $i, j = 1, 2, \dots, n$.

Allo stesso modo possiamo ottenere la derivata di J rispetto a b_{ij}

$$\begin{aligned} \frac{\partial J}{\partial b_{ij}} &= - \sum_{t=1}^N (d(t) - y(t)) \frac{\partial y(t)}{\partial b_{ij}} \\ &= - \sum_{t=1}^N (d(t) - y(t)) \sum_{k=1}^n c_k \frac{\partial z_k}{\partial b_{ij}} \end{aligned} \quad (3.36)$$

Definito $\phi_{ij}(t) \triangleq \frac{\partial \mathbf{z}(t)}{\partial b_{ij}}$, con un procedimento analogo al precedente, otteniamo

$$\phi_{ij}(t) = \Lambda(\mathbf{x}(t))[A\phi_{ij}(t-1) + Q_{ij}\mathbf{z}(t-1)] \quad (3.37)$$

Abbiamo in questo caso

$$\frac{\partial J}{\partial b_{ij}} = - \sum_{t=1}^N (d(t) - y(t)) \mathbf{c}^T \phi_{ij}(t) \quad (3.38)$$

per $i = 1, \dots, n$ e $j = 1, \dots, m$. In più abbiamo

$$\frac{\partial J}{\partial b_{0j}} = - \sum_{t=1}^N (d(t) - y(t)) \mathbf{c}^T \psi_j^0(t) \quad (3.39)$$

dove b_{0j} è l'elemento j -esimo del vettore \mathbf{b}_0 e

$$\phi_j^0(t) = \Lambda(\mathbf{x}(t))[A\psi_j^0(t-1)] \quad (3.40)$$

Una volta ottenute le derivate di J rispetto ai parametri, questi possono essere aggiornati con l'algoritmo del gradiente discendente.

Termine di momento

È possibile includere un termine di momento α nell'algoritmo di aggiornamento dei pesi nel seguente modo:

$$\Delta\hat{\theta}_{k+1} = -\eta J'(\hat{\theta}) + \alpha\Delta\hat{\theta}_k \quad (3.41)$$

La funzione di questo termine è di fornire informazioni del secondo ordine nell'algoritmo di apprendimento. In questo modo si ha una più veloce convergenza nelle regioni piatte, dove il gradiente è piccolo, ma oscillazioni nelle regioni strette della superficie di errore nello spazio dei pesi.

3.4.3 Metodi di ottimizzazione

In questo paragrafo analizzeremo tecniche di minimizzazione basate sui metodi di Newton o su una loro modificazione, per incorporare più informazione negli algoritmi di apprendimento. Si chiamano metodi "quasi" del secondo ordine perchè approssimano i metodi del secondo ordine, usando informazione basata sulla derivata del primo ordine della funzione di errore.

Gradiente coniugato

Un problema nell'ottimizzazione numerica è che le direzioni di discesa del gradiente interferiscono l'una con l'altra cosicché, una buona discesa nello step precedente può essere vanificata da una discesa nello step successivo in direzione completamente differente. Questo può essere evitato usando il gradiente coniugato:

1. Si selezionano un valore iniziale del vettore θ_0 ed una serie di direzioni coniugate

$$\mathbf{r}_0 = -J'(\theta_0) \quad (3.42)$$

$$\mathbf{p}_0 = \mathbf{r}_0 \quad (3.43)$$

2. Si trova il minimo, per esempio con il metodo del line search, della seguente espressione

$$\alpha_k = \min_{\alpha} J(\theta_k + \alpha \mathbf{p}_k) \quad (3.44)$$

3. Si aggiornano i pesi come segue

$$\theta_{k+1} = \theta_k + \alpha_k \mathbf{p}_k \quad (3.45)$$

$$\mathbf{r}_{k+1} = -J'(\theta_{k+1}) \quad (3.46)$$

4. Se $k \bmod n_{\theta} = 0$, allora

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} \quad (3.47)$$

altrimenti

$$\beta_k = \frac{|\mathbf{r}_{k+1}|^2 - \mathbf{r}_{k+1}^T \mathbf{r}_k}{|\mathbf{r}_k|^2} \quad (3.48)$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} - \beta_k \mathbf{p}_k \quad (3.49)$$

5. $k = k + 1$, oppure termina se è soddisfatto qualche criterio di stop.

Il gradiente coniugato ha una maggiore velocità di convergenza rispetto ai metodi del primo ordine. Il calcolo del gradiente, che compare negli step 1 e 3 può essere effettuato con il metodo esposto nel paragrafo precedente.

Gradiente coniugato scalato

Nell'algoritmo del gradiente coniugato dobbiamo effettuare un "line search" per calcolare il valore di α_k . Questa è una operazione computazionalmente abbastanza

onerosa. Nell'algoritmo del gradiente coniugato scalato l'idea essenziale, per ovviare a questo problema, è sostituire l'Hessiano della funzione di errore con il termine $J''(\theta) + \lambda_k I$. A questo punto il learning rate può essere calcolato nel seguente modo:

$$\alpha_k = \frac{-J'(\theta_k)\mathbf{p}_k}{\mathbf{p}_k^T J''(\theta_k)\mathbf{p}_k + \lambda_k |\mathbf{p}_k|^2} \quad (3.50)$$

Il valore di λ_k viene incrementato o decrementato in ogni iterazione a seconda di quanto è buona l'approssimazione del secondo ordine. La bontà dell'approssimazione viene calcolata dalla seguente espressione, che rappresenta il rapporto tra la variazione dell'errore reale e la variazione dell'errore quadratico di predizione:

$$\Delta_k = \frac{(\mathbf{p}_k^T J''(\theta_k)\mathbf{p}_k + \lambda_k |\mathbf{p}_k|^2) (J(\theta_k) - J(\theta_{k+1}))}{(-J'(\theta_k)^T \mathbf{p}_k)^2} \quad (3.51)$$

Quindi l'incremento di λ_k è dipendente da Δ_k .

Inoltre il prodotto fra l'Hessiana e \mathbf{p}_k può approssimarsi con

$$J''(\theta_k)\mathbf{p}_k \approx \frac{J'(\theta_k + \sigma_k \mathbf{p}_k) - J'(\theta_k)}{\sigma_k} \quad (3.52)$$

dove σ_k è data da

$$\sigma_k = \frac{\varepsilon}{|\mathbf{p}_k|^2} \quad (3.53)$$

con $0 < \varepsilon < 1$. A questo punto dobbiamo valutare solo i gradienti con il metodo analizzato nella sezione precedente.

3.5 Apprendimento e generalizzazione

Dobbiamo ora puntualizzare che lo scopo dell'addestramento della rete non è apprendere una esatta rappresentazione del training data, ma costruire un modello statistico del processo che genera i dati. Questo è fondamentale se vogliamo che la rete mostri delle buone capacità di generalizzazione, in modo da predire bene i nuovi dati. Si può per analogia considerare il problema del fitting polinomiale: se usiamo

un polinomio di grado troppo basso (underfitting) si ha una cattiva rappresentazione dei dati ed una cattiva predizione perchè la funzione che usiamo è poco flessibile (vedi figura 3.10). D'altra parte, se usiamo troppi parametri, si ottiene un aumento

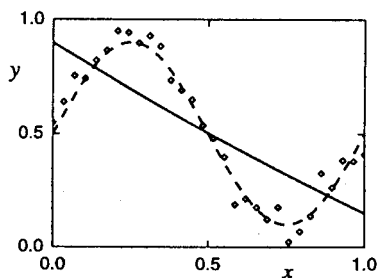


Figura 3.10: Esempio di underfitting polinomiale

della complessità del modello, una cattiva generalizzazione ed una rappresentazione sbagliata del sistema fisico che genera i dati. Principalmente la stima dei parametri è troppo condizionata dal rumore presente nel training set (vedi figura 3.11). Il numero di coefficienti del polinomio controlla la complessità del modello. Appare allora chiara l'importanza di ottimizzare il numero di parametri, in modo da non rendere il modello troppo complesso ed ottenere la massima generalizzazione e la migliore interpolazione (vedi figura 3.12).

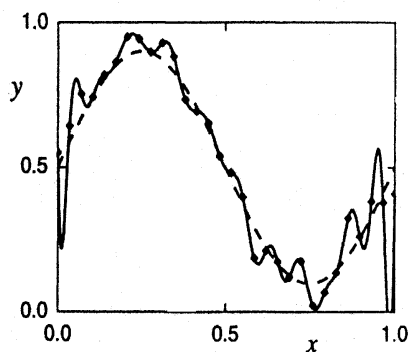


Figura 3.11: Esempio di overfitting polinomiale

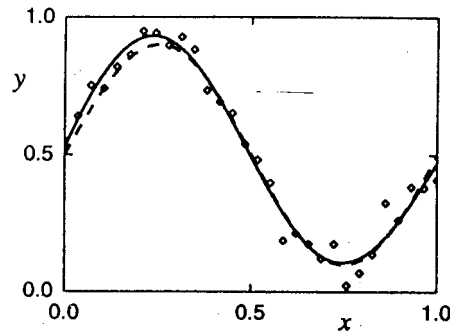


Figura 3.12: Esempio di fitting corretto

Nel caso delle reti neurali, la complessità può essere controllata cambiando l'architettura della rete, cioè il numero di parametri adattivi. Un modo per implementare questa procedura è confrontare le performance di diverse reti con numero di neuroni hidden differente. Oppure si può addestrare una rete relativamente grande e poi eliminare le connessioni o le unità meno significative. Similmente si può partire con reti con poche unità hidden per poi aggiungerle nel processo di apprendimento per ottenere la struttura ottima.

Un secondo approccio consiste nel controllare la complessità aggiungendo un termine di *regolarizzazione* nella funzione di errore. Infine, nelle applicazioni pratiche, dobbiamo ottimizzare la complessità del modello, dato un certo training set. Una delle tecniche più importanti, in questo caso è chiamata *cross-validation*.

3.5.1 Regolarizzazione

La tecnica di regolarizzazione consiste nel definire una nuova funzione di errore del tipo:

$$\tilde{E} = E + \nu\Omega \quad (3.54)$$

dove E è la funzione di errore standard, ν è un parametro di controllo e Ω è il termine di regolarizzazione. Una funzione $y(x)$, per fittare bene i dati di training

deve avere un valore basso di E e per essere smooth un valore basso di Ω . Il risultato è un compromesso fra l'interpolazione dei dati e la minimizzazione di Ω .

Vi sono diversi tipi di termini di regolarizzazione, tutti appartenenti alla classe dei regolarizzatori di *Tikhonov*:

$$\Omega = \frac{1}{2} \sum_{r=0}^R \int_a^b h_r(x) \left(\frac{d^r y}{dx^r} \right)^2 dx \quad (3.55)$$

dove $h_r \geq 0$ per $r = 0, \dots, R-1$, e $h_R > 0$.

Decadimento dei pesi

Una delle forme più semplici dei termini di regolarizzazione è il *decadimento dei pesi*, e consiste nella somma quadratica dei parametri adattivi della rete:

$$\Omega = \frac{1}{2} \sum_i w_i^2 \quad (3.56)$$

dove la somma è estesa a tutti i pesi e le soglie. È stato provato in modo empirico che un termine di regolarizzazione di questa forma migliora significativamente la capacità di generalizzazione della rete.

Molti algoritmi di addestramento usano la derivata dell'errore rispetto ai pesi. Richiamando l'equazione 3.54

$$\nabla \tilde{E} = \nabla E + \nu \mathbf{w} \quad (3.57)$$

Se consideriamo la semplice regola del gradiente discendente ed omettiamo il termine E , si ottiene la legge di evoluzione nel limite di tempo continuo:

$$\frac{d\mathbf{w}(t)}{dt} = -\eta \nabla = -\eta \nu \mathbf{w} \quad (3.58)$$

dove η è il learning rate. La soluzione è

$$\mathbf{w}(t) = \mathbf{w}(0) \exp(-\eta \nu t) \quad (3.59)$$

cosicché tutti i pesi decadono esponenzialmente a zero. Questo è il motivo del termine "decadimento dei pesi".

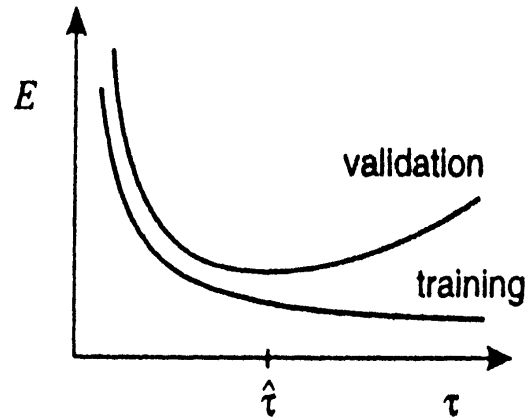


Figura 3.13: Esempio schematico dell'uso del validation set

Early stopping

Un modo alternativo di regolarizzazione per controllare la effettiva complessità del modello è la procedura di *early-stopping*. L'addestramento di una rete non lineare consiste nella riduzione iterativa della funzione di errore. Durante una tipica sessione di addestramento, per molti algoritmi come il gradiente coniugato scalato, l'errore è una funzione monotona decrescente dell'indice di iterazione. Tuttavia, calcolando l'errore misurato rispetto ad un altro insieme di dati **indipendente** dal training set, detto *validation set*, spesso si nota prima un decremento, seguito da un incremento quando la rete incomincia l'overfit (vedi figura 3.13). L'addestramento può essere fermato quando si verifica questa risalita dell'errore relativo al validation set, in modo da ottenere la migliore performance di generalizzazione.

3.6 Preprocessing

Poiché le reti neurali possono in principio approssimare qualsiasi relazione tra due variabili, possiamo implementare direttamente una rete che ci dia la rappresentazione fra due set di dati. In realtà per molte applicazioni è necessario prima

trasformare i dati in una nuova rappresentazione e poi effettuare il training. Questa fase, detta *preprocessing*, è la più importante nell'analisi dei dati, ed è essenziale affinché la rete dia buoni risultati. Vedremo nei successivi paragrafi le principali tecniche di preprocessing.

3.6.1 Normalizzazione e whitening

Una delle più comuni forme di preprocessing consiste semplicemente in una normalizzazione delle variabili di input, nel caso in cui i valori siano significativamente differenti. Per fare questo trattiamo ogni variabile x_i indipendentemente, e ne calcoliamo media e varianza nel seguente modo

$$\bar{x}_i = \frac{1}{N} \sum_{i=1}^N x_i^n \quad (3.60)$$

$$\sigma_i = \frac{1}{N-1} \sum_{i=1}^N (x_i^n - \bar{x}_i)^2 \quad (3.61)$$

dove $n = 1, \dots, N$ sono gli indici dei patterns. Ora possiamo definire un nuovo set di patterns normalizzati

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i} \quad (3.62)$$

che hanno media zero e varianza unitaria. In questo modo siamo sicuri che i valori delle variabili sono tutti dello stesso ordine di grandezza.

Una normalizzazione più sofisticata, che considera anche la correlazione fra le variabili, è il *whitening*. Raggruppiamo le variabili in un vettore $\mathbf{x} = (x_1, \dots, x_d)$ con media e matrice di covarianza

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n \\ \Sigma &= \frac{1}{N-1} \sum_{n=1}^N (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T \end{aligned} \quad (3.63)$$

Se introduciamo l'equazione agli autovalori

$$\mathbf{\Sigma} \mathbf{u}_j = \lambda_j \mathbf{u}_j \quad (3.64)$$

otteniamo un nuovo gruppo di variabili trasformate linearmente

$$\tilde{\mathbf{x}}^n = \mathbf{\Lambda}^{1/2} \mathbf{U}^T (\mathbf{x}^n - \bar{\mathbf{x}}) \quad (3.65)$$

dove abbiamo definito

$$\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_d) \quad (3.66)$$

$$\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d) \quad (3.67)$$

È facile verificare che in questo modo il data set ha media nulla e matrice di covarianza unitaria, come si nota in figura 3.14.

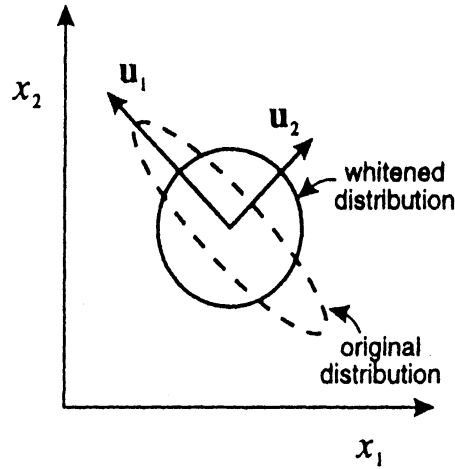


Figura 3.14: Risultato della operazione di whitening

3.6.2 Principal Component Analysis

La Principal Component Analysis, o PCA, è usata intensamente nel signal processing, nella statistica e nel neural computing. In altre aree prende anche il nome di

trasformata discreta di Karhunen-Loève o trasformata di Hotelling. Compito principale della PCA è effettuare una trasformazione, lineare in prima approssimazione, per ridurre la dimensione dello spazio degli input.

L'idea di base è mappare un vettore \mathbf{x}^n di un spazio d -dimensionale in un vettore \mathbf{z}^n di uno spazio M -dimensionale, con $d > M$. Notiamo subito che qualsiasi vettore \mathbf{x} può essere espresso come combinazione lineare di una base $\{\mathbf{u}\}$ in \mathbb{R}^d .

$$\mathbf{x} = \sum_{i=1}^d z_i \mathbf{u}_i \quad (3.68)$$

dove le \mathbf{u}_i soddisfano la relazione di ortogonalità

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \quad (3.69)$$

ed i coefficienti z_i si ricavano da

$$z_i = \mathbf{u}_i^T \mathbf{x} \quad (3.70)$$

che corrisponde ad una rotazione del sistema di coordinate da un sistema di assi dati dalle \mathbf{x} ad un nuovo sistema di coordinate dato dalle z_i .

Supponiamo di voler utilizzare solo M vettori di base e quindi solo M coefficienti z_i . Otteniamo l'espressione approssimata

$$\tilde{\mathbf{x}} = \sum_{i=1}^M z_i \mathbf{u}_i + \sum_{i=M+1}^d b_i \mathbf{u}_i \quad (3.71)$$

dove b_i sono costanti. Consideriamo tutto il data set di N vettori \mathbf{x}^n . Dobbiamo scegliere \mathbf{u}_i e b_i cosicché l'approssimazione data dalla equazione 3.71, con i valori z_i ricavati dalla relazione 3.70, sia la migliore, mediata su tutto il data set.

L'errore dovuto all'approssimazione risulta

$$\mathbf{x}^n - \tilde{\mathbf{x}}^n = \sum_{i=M+1}^d (z_i^n - b_i) \mathbf{u}_i \quad (3.72)$$

Possiamo definire l'errore quadratico medio

$$E_M = \frac{1}{2} \sum_{n=1}^N \|\mathbf{x}^n - \tilde{\mathbf{x}}^n\|^2 = \frac{1}{2} \sum_{n=1}^N \sum_{i=M+1}^d (z_i^n - b_i)^2 \quad (3.73)$$

Per minimizzare E_M dobbiamo risolvere $\partial E_M / \partial b_i = 0$. Otteniamo

$$b_i = \frac{1}{N} \sum_{n=1}^N z_i^n = \mathbf{u}_i \bar{\mathbf{x}} \quad (3.74)$$

definito

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n \quad (3.75)$$

Usando la relazione 3.70 e la 3.74, possiamo scrivere l'errore quadratico medio come

$$\begin{aligned} E_M &= \frac{1}{2} \sum_{n=1}^N \sum_{i=M+1}^d \{ \mathbf{u}_i^T (\mathbf{x}^n - \bar{\mathbf{x}}) \}^2 \\ &= \frac{1}{2} \sum_{i=M+1}^d \mathbf{u}_i^T \Sigma \mathbf{u}_i \end{aligned} \quad (3.76)$$

dove Σ è la matrice di covarianza del set di vettori $\{\mathbf{x}^n\}$ data da

$$\Sigma = \sum_n (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T \quad (3.77)$$

Dobbiamo ora minimizzare E_M rispetto ai vettori di base \mathbf{u}_i . Si può mostrare che questo corrisponde a scegliere gli \mathbf{u}_i pari agli autovettori della matrice di covarianza.

$$\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (3.78)$$

Il minimo dell'errore quadratico medio si ha per

$$E_M = \frac{1}{2} \sum_{i=M+1}^d \lambda_i \quad (3.79)$$

Quindi il minimo viene raggiunto scegliendo i $d - M$ autovalori più piccoli ed i loro corrispondenti autovettori.

In pratica l'algoritmo si realizza nel seguente modo:

1. si centra il data set \mathbf{x}^n intorno alla sua media.
2. si calcola la matrice di covarianza

3. si ricavano gli autovalori e gli autovettori
4. gli autovettori corrispondenti ai M autovalori più grandi generano il sottospazio su cui vengono proiettati i vettori \mathbf{x}^n .
5. l'errore di approssimazione viene valutato tramite l'equazione 3.79.

In figura 3.15 possiamo vedere un esempio della prima componente principale, su cui poter proiettare i dati per un set bidimensionale.

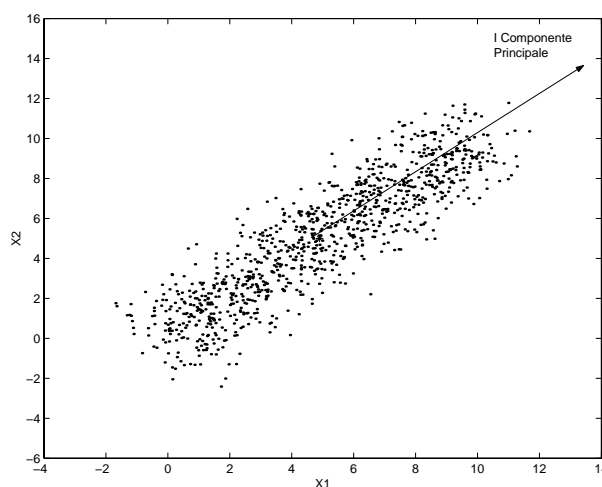


Figura 3.15: Esempio di estrazione della prima componente principale

3.7 Bayesian learning

Questa parte della tesi sarà dedicata allo studio della inferenza Bayesiana per l'addestramento delle reti neurali [6], [7], [8]. I metodi Bayesiani offrono diversi vantaggi rispetto alle tecniche analizzate precedentemente:

- I metodi convenzionali di minimizzazione dell'errore sono una particolare approssimazione dell'approccio Bayesiano.

- La complessità del modello può essere facilmente controllata.
- Per i problemi di regressione, gli intervalli di confidenza possono essere facilmente assegnati all'output della rete.
- Non è necessario un ulteriore set di validation.
- I differenti modelli possono essere paragonati solo usando il training set.
- I pesi statistici dei differenti input possono essere valutati tramite dei coefficienti di regolarizzazione differenti per ogni input. Se il valore di tale coefficiente è alto, l'input relativo è poco importante per la determinazione dell'output e può essere eliminato.

Uno degli aspetti più interessanti dell'approccio Bayesiano è il modo in cui viene trattata la complessità del modello. Consideriamo tre ipotetici modelli, $\mathcal{H}_1, \mathcal{H}_2$ e \mathcal{H}_3 , con complessità crescente. Ogni modello ha la propria architettura ed è regolato da una serie di parametri adattivi. Al variare di questi parametri ogni modello rappresenta un range di funzioni che legano l'input all'output della rete. Sia $D \equiv (\mathbf{t}^1, \dots, \mathbf{t}^N)$ il set di vettori target. Dal teorema di Bayes possiamo calcolare la probabilità a posteriori per ogni modello:

$$p(\mathcal{H}_i|D) = \frac{p(D|\mathcal{H}_i)p(\mathcal{H}_i)}{p(D)} \quad (3.80)$$

La quantità $p(\mathcal{H}_i)$ rappresenta la probabilità a priori. Poichè non c'è alcun motivo per preferire un modello ad un altro, può essere scelta uguale per ogni modello \mathcal{H}_i . Il denominatore $p(D)$ è indipendente da \mathcal{H}_i e quindi i vari modelli possono essere comparati valutando $p(D|\mathcal{H}_i)$, detta *evidenza statistica*. Quindi l'approccio Bayesiano può essere usato per scegliere il modello con evidenza maggiore, come si vede in figura 3.16.

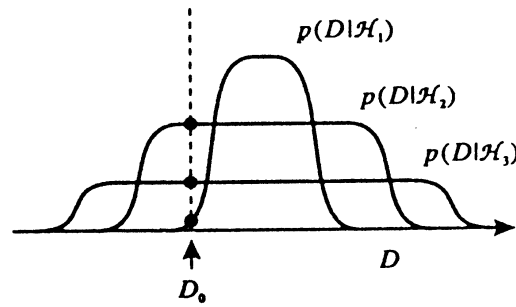


Figura 3.16: Selezione Bayesiana del modello con dataset D_0

Ma possiamo anche considerare tutti i modelli ed ottenere un output dato da una somma della risposta dei vari modelli pesata in modo proporzionale all'evidenza. Così il modello più probabile contribuisce maggiormente alla determinazione dell'output della rete [10], [11].

3.7.1 Apprendimento Bayesiano dei pesi della rete

Il primo problema è come determinare i pesi della rete dal training set. Abbiamo visto fino ad ora tecniche basate sulla massima verosimiglianza. L'apprendimento Bayesiano considera invece una funzione di distribuzione sullo spazio dei pesi. Inizialmente questa funzione viene scelta uniforme. Successivamente, con il training set, si possono ottenere, tramite il teorema di Bayes, le probabilità a posteriori. Infine queste possono essere usate per predire nuovi valori con la rete addestrata.

Distribuzione dei pesi

Consideriamo una rete la cui architettura è fissa. Nell'apprendimento Bayesiano consideriamo una distribuzione di probabilità sui pesi. In assenza di alcun dato questa distribuzione è descritta da una funzione a priori $p(\mathbf{w})$, in cui $\mathbf{w} \equiv (w_1, \dots, w_W)$ sono i parametri da adattare. Sia $D \equiv (\mathbf{t}^1, \dots, \mathbf{t}^N)$ l'insieme dei vettori target.

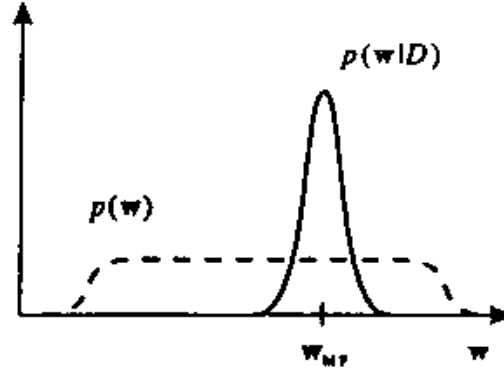


Figura 3.17: Inferenza Bayesiana dei pesi della rete

Tramite il teorema di Bayes possiamo scrivere:

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \quad (3.81)$$

dove il denominatore è un fattore di normalizzazione dato da

$$p(D) = \int p(D|\mathbf{w})p(\mathbf{w})d\mathbf{w} \quad (3.82)$$

Possiamo includere anche l'insieme dei vettori di input $\chi \equiv (\mathbf{x}^1, \dots, \mathbf{x}^N)$ nel teorema di Bayes e scrivere

$$p(\mathbf{w}|D, \chi) = \frac{p(D|\mathbf{w}, \chi)p(\mathbf{w}|\chi)}{p(D|\chi)} \quad (3.83)$$

Il formalismo Bayesiano per l'apprendimento è dunque il seguente. Si sceglie una distribuzione a priori dei pesi $p(\mathbf{w})$ abbastanza piatta, in modo da non limitare il valore che i parametri possono assumere. Una volta osservati i dati, possiamo convertire questa distribuzione a priori in una distribuzione a posteriori usando il teorema di Bayes. Questa nuova funzione sarà più stretta e piccata verso il valore \mathbf{w}_{MP} , il che indica che è stato appreso il modello statistico di generazione dei dati (vedi figura 3.17). Per valutare la distribuzione a posteriori dobbiamo trovare una distribuzione a priori $p(\mathbf{w})$ e la funzione di verosimiglianza $p(D|\mathbf{w})$.

Distribuzione a priori gaussiana

La distribuzione a priori dei pesi riflette le conoscenze che abbiamo sulla funzione che stiamo cercando. In generale possiamo sceglierla del tipo

$$p(\mathbf{w}) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) \quad (3.84)$$

dove $Z_W(\alpha)$ è una fattore di normalizzazione dato da

$$Z_W(\alpha) = \int \exp(-\alpha E_W) d\mathbf{w} \quad (3.85)$$

Per favorire piccoli valori dei pesi che garantiscano una buona generalizzazione possiamo scegliere:

$$E_W = \frac{1}{2} \sum_{i=1}^W w_i^2 \quad (3.86)$$

che corrisponde ad una distribuzione a priori di pesi della rete della forma:

$$p(\mathbf{w}) = \left(\frac{\alpha}{2\pi}\right)^{W/2} \exp\left(-\frac{\alpha}{2} \|\mathbf{w}\|^2\right) \quad (3.87)$$

Il parametro α è chiamato *iperparametro* ed è parte dal processo di apprendimento.

Il termine di normalizzazione può essere facilmente calcolato imponendo

$$\int p(\mathbf{w}) d\mathbf{w} = 1$$

e si ottiene

$$Z_W(\alpha) = \left(\frac{2\pi}{\alpha}\right)^{W/2} \quad (3.88)$$

Modello di rumore gaussiano

Consideriamo ora un'architettura feedforward generale per il problema della regressione. In generale possiamo scrivere la funzione di verosimiglianza come

$$p(D|\mathbf{w}) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \quad (3.89)$$

dove β è un altro iperparametro, E_D una funzione di errore e $Z_D(\beta)$ è un fattore di normalizzazione.

Assumendo che il valore di target è generato da una funzione regolare con rumore gaussiano additivo a media zero e varianza unitaria, la probabilità di osservare un dato valore t per un dato input \mathbf{x} è

$$p(t|\mathbf{x}, \mathbf{w}) \propto \exp\left(-\frac{\beta}{2}\{y(\mathbf{x}; \mathbf{w}) - t\}^2\right) \quad (3.90)$$

dove $y(\mathbf{x}; \mathbf{w})$ è l'output della rete e l'iperparametro β controlla la varianza del rumore. Per l'indipendenza dei dati da questa distribuzione possiamo scrivere

$$\begin{aligned} p(D|\mathbf{w}) &= \prod_{n=1}^N p(t^n|\mathbf{x}^n, \mathbf{w}) \\ &= \frac{1}{Z_D(\beta)} \exp\left(-\frac{\beta}{2} \sum_{n=1}^N \{y(\mathbf{x}^n; \mathbf{w}) - t^n\}^2\right) \end{aligned} \quad (3.91)$$

dove $Z_D(\beta) = (2\pi/\beta)^{N/2}$. β lo abbiamo considerato costante, ma può essere determinato nella fase di apprendimento.

Distribuzione a posteriori dei pesi della rete

Una volta scelta la distribuzione a priori e la funzione di verosimiglianza, possiamo usare il teorema di Bayes 3.81 per calcolare la distribuzione a posteriori dei pesi, ottenendo:

$$p(\mathbf{w}|D) = \frac{1}{Z_S} \exp(-\beta E_D - \alpha E_W) = \frac{1}{Z_S} \exp(-S(\mathbf{w})) \quad (3.92)$$

dove

$$s(\mathbf{w}) = \beta E_D + \alpha E_W \quad (3.93)$$

e

$$Z_S(\alpha, \beta) = \int \exp(-\beta E_D - \alpha E_W) d\mathbf{w} \quad (3.94)$$

Per determinare il valore \mathbf{w}_{ott} che massimizza la probabilità a posteriori, dobbiamo minimizzare il logaritmo negativo della 3.92. Poiché il fattore di normalizzazione Z_S non dipende dai pesi, possiamo minimizzare direttamente $S(\mathbf{w})$. Data la scelta della distribuzione a priori e del modello di rumore, l'equazione 3.93 può essere scritta come:

$$S(\mathbf{w}) = \frac{\beta}{2} \sum_{n=1}^N \{y(\mathbf{x}^n; \mathbf{w}) - t^n\}^2 + \frac{\alpha}{2} \sum_{i=1}^W w_i^2 \quad (3.95)$$

Il valore che massimizza la probabilità a posteriori corrisponde al minimo della 3.95.

Consistenza a priori

Abbiamo visto che scegliere una distribuzione dei pesi a priori di tipo quadratico, corrisponde ad una semplice regolarizzazione di decadimento delle connessioni, bias incluso. Possiamo estendere il concetto e considerare un coefficiente di regolarizzazione per ogni strato di connessioni, incluse le soglie.

Per una rete a due strati possiamo scegliere una distribuzione del tipo

$$p(\mathbf{w}) \propto \exp \left(-\frac{\alpha_1}{2} \sum_{w \in \mathcal{W}_1} w^2 - \frac{\alpha_2}{2} \sum_{w \in \mathcal{W}_2} w^2 \right) \quad (3.96)$$

dove \mathcal{W}_1 denota il set di pesi del primo strato e \mathcal{W}_2 il set del secondo strato. In generale possiamo considerare il set di pesi divisi in k classi \mathcal{W}_k e scegliere la distribuzione a priori come:

$$p(\mathbf{w}) \propto \exp \left(-\frac{1}{2} \sum_k \alpha_k \|\mathbf{w}\|_k^2 \right) \quad (3.97)$$

con

$$\|\mathbf{w}\|_k^2 = \sum_{w \in \mathcal{W}_k} w^2 \quad (3.98)$$

Un schema molto utilizzato è quello introdotto da Mackay detto *ARD model* (Automatic Relevance Determination): per una rete con un solo livello hidden si

utilizza una classe per ogni input, cioè il peso dall'input i -esimo allo strato hidden, una classe per le connessioni dal bias allo strato hidden, ed una classe per ogni output, costituita dal bias e da tutti i pesi dallo strato hidden.

3.7.2 Distribuzione degli output della rete e barre di errore

Abbiamo visto che, nel formalismo Bayesiano, una rete addestrata è descritta da una distribuzione di probabilità a posteriori dei pesi. Presentato un nuovo input alla rete, allora la distribuzione dei pesi induce una distribuzione degli output della rete dipendente dal dato input.

Possiamo allora scrivere la distribuzione dell'output della rete per un dato input \mathbf{x} come

$$p(t|\mathbf{x}, D) = \int p(t|\mathbf{x}, \mathbf{w})p(\mathbf{w}|D)d\mathbf{w} \quad (3.99)$$

dove $p(\mathbf{w}|D)$ è la distribuzione a posteriori dei pesi. La distribuzione $p(t|\mathbf{x}, \mathbf{w})$ è semplicemente il modello della distribuzione del rumore sui target, per un dato valore dei pesi, ed è data da

$$p(t|\mathbf{x}, \mathbf{w}) \propto \exp\left(-\frac{\beta}{2}\{y(\mathbf{x}; \mathbf{w}) - t\}^2\right) \quad (3.100)$$

Utilizzando l'approssimazione gaussiana per la distribuzione a posteriori dei pesi, possiamo valutare la precedente espressione, ed ottenere:

$$p(t|\mathbf{x}, D) \propto \int \exp\left(-\frac{\beta}{2}\{t - y(\mathbf{x}; \mathbf{w})\}^2\right) \exp\left(-\frac{1}{2}\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\right) d\mathbf{w} \quad (3.101)$$

dove abbiamo tralasciato i termini indipendenti da t . Possiamo poi assumere che la larghezza della funzione di distribuzione a posteriori, determinata dall'Hessiana \mathbf{A} , è sufficientemente stretta cosicché la funzione $y(\mathbf{x}; \mathbf{w})$ può essere scritta in serie intorno al punto \mathbf{w}_{MP} :

$$y(\mathbf{x}; \mathbf{w}) \simeq y(\mathbf{x}; \mathbf{w}_{MP}) + \mathbf{g}^T \Delta\mathbf{w} \quad \text{con} \quad \mathbf{g} \equiv \nabla_{\mathbf{w}} y|_{\mathbf{w}_{MP}} \quad (3.102)$$

Ora l'integrale è facilmente valutabile e si ottiene

$$p(t|\mathbf{x}, D) = \frac{1}{(2\pi\sigma_t^2)^{1/2}} \exp\left(-\frac{(t - y(\mathbf{x}; \mathbf{w}_{MP}))^2}{2\sigma_t^2}\right) \quad (3.103)$$

Questa distribuzione ha una media data da y_{MP} ed una varianza

$$\sigma_t^2 = \frac{1}{\beta} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g} \quad (3.104)$$

Possiamo interpretare la deviazione standard σ_t^2 della distribuzione prevista per t come barra di errore sul valor medio y_{MP} . L'error bar ha due contributi: uno proviene dal rumore intrinseco sui dati di target, corrispondente al primo termine della 3.104, ed uno proviene dalla varianza della funzione di distribuzione a posteriori dei pesi della rete, corrispondente al secondo termine della 3.104. Si nota allora come l'approccio Bayesiano all'apprendimento permette il calcolo delle error bar sull'output della rete. In pratica prima si calcola il valore più probabile per i pesi \mathbf{w}_{MP} minimizzando la funzione regolarizzata $S(\mathbf{w})$. Possiamo poi associare le error bar alla funzione della rete $y(\mathbf{x}; \mathbf{w})$ valutando l'Hessiana \mathbf{A} ed usando l'equazione 3.104.

3.7.3 Integrazione sugli iperparametri α e β

Abbiamo assunto, fino ad ora, noti i valori degli iperparametri α e β . In genere abbiamo solo una stima dell'intervallo di variazione dei parametri, ed in particolare una idea del livello di rumore β . Il modo corretto, dal punto di vista bayesiano, per valutare gli iperparametri non noti è integrarli nelle distribuzioni a posteriori. Per esempio, la distribuzione a posteriori dei pesi della rete è data da

$$\begin{aligned} p(\mathbf{x}|D) &= \int \int p(\mathbf{w}, \alpha, \beta|D) d\alpha d\beta \\ &= \int \int p(\mathbf{w}|\alpha, \beta, D) d\alpha d\beta \end{aligned} \quad (3.105)$$

Abbiamo praticamente esteso la nostra notazione in modo da rendere esplicita la dipendenza dagli iperparametri nelle varie funzione di distribuzioni.

I due approcci principali per l'integrazione sono *l'approssimazione dell'evidenza* discussa MacKay (1992), e *l'integrazione analitica*, che noi analizzeremo in dettaglio. Scriviamo l'integrale 3.105 nella forma:

$$\begin{aligned} p(\mathbf{x}|D) &= \int \int p(\mathbf{w}, \alpha, \beta | D) d\alpha d\beta \\ &= \frac{1}{p(D)} \int \int p(D|\mathbf{w}, \beta) p(\mathbf{w}|\alpha) p(\alpha) p(\beta) d\alpha d\beta \end{aligned} \quad (3.106)$$

dove abbiamo usato il teorema di Bayes nella forma descritta dall'equazione 3.81, ed il fatto che $p(D|\mathbf{w}, \alpha, \beta) = p(D|\mathbf{w}, \beta)$ perchè il termine di verosimiglianza è indipendente da α . Similmente $p(\mathbf{w}|\alpha, \beta) = p(\mathbf{w}|\alpha)$ perchè la distribuzione a priori dei pesi è indipendente da β . Infine, nell'assunzione dell'indipendenza statistica degli iperparametri, $p(\alpha, \beta) = p(\alpha)p(\beta)$.

Scegliamo una distribuzione a priori degli iperparametri uniforme in scala logaritmica:

$$p(\ln \alpha) = \frac{1}{\alpha} \quad p(\ln \beta) = \frac{1}{\beta} \quad (3.107)$$

Consideriamo l'integrale in $d\alpha$ definito dall'equazione 3.106. Utilizzando l'equazione 3.84 abbiamo

$$\begin{aligned} p(\mathbf{w}) &= \int_0^\infty p(\mathbf{w}|\alpha) p(\alpha) d\alpha \\ &= \int_0^\infty \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) \frac{1}{\alpha} d\alpha \\ &= (2\pi)^{-W/2} \int_0^\infty \exp(-\alpha E_W) \alpha^{(W/2)-1} d\alpha \\ &= \frac{\Gamma(W/2)}{(2\pi E_W)^{W/2}} \end{aligned} \quad (3.108)$$

dove Γ è la funzione Gamma standard. L'integrazione su β può essere effettuata allo stesso modo e risulta

$$p(D|\mathbf{w}) = \frac{\Gamma(N/2)}{(2\pi E_D)^{N/2}} \quad (3.109)$$

Possiamo ora scrivere esattamente la distribuzione a posteriori non normalizzata dei pesi. Il logaritmo negativo di tale distribuzione rappresenta la funzione di errore:

$$-\ln p(\mathbf{w}|D) = \frac{N}{2} \ln E_D + \frac{W}{2} \ln E_W + cost \quad (3.110)$$

che può essere riscritta come

$$-\ln p(\mathbf{w}|D) = \beta E_D + \alpha E_W + cost \quad (3.111)$$

il cui gradiente è

$$-\nabla \ln p(\mathbf{w}|D) = \beta \nabla E_D + \alpha \nabla E_W \quad (3.112)$$

Definiti i valori di

$$\begin{aligned} \alpha_{eff} &= W/2E_W \\ \beta_{eff} &= N/2E_D \end{aligned} \quad (3.113)$$

Il gradiente può esprimersi come

$$-\nabla \ln p(\mathbf{w}|D) = \beta_{eff} \nabla E_D + \alpha_{eff} \nabla E_W \quad (3.114)$$

Allora la minimizzazione della funzione di errore 3.110 può essere implementata minimizzando la 3.111 nella quale i valori di α_{eff} e β_{eff} stimati in modo adattivo utilizzando le formule 3.113.

3.7.4 Implementazione pratica

Dopo aver analizzato i metodi Bayesiani per la fase di learning della rete, riassumiamo i passi principali per implementare tali tecniche.

1. Scegliere i valori iniziali degli iperparametri β ed α_k , $\forall \mathcal{W}_k$, e dei pesi campionati da una qualche distribuzione a priori (in genere uniforme).
2. Minimizzare la funzione di errore totale $S(\mathbf{w})$ con qualche criterio di ottimizzazione.

3. Dopo qualche ciclo dell'algoritmo, stimare nuovamente i valori di α e β usando le formule 3.113.
4. Ripetere i passi 1-3 finché non si è raggiunto un criterio di stop.
5. Ripetere i passi 1-4 per diversi valori iniziali dei parametri per evitare minimi locali.

Bibliografia relativa al capitolo III

- [1] C. M. Bishop, *Neural Networks for Pattern Recognition*, Claredon Press - Oxford, 1997
- [2] A. C. Tsoi e A. Back, *Discrete time recurrent neural network architectures: A unifying review*, *Neurocomputing* 15 (1997) 183-223
- [3] A. C. Tsoi, *Recurrent Neural Network Architectures: An overview in Adaptive Processing of Sequences and Data Structures*, Proceedings della "International Summer School on Neural Networks E.R. Caianiello", Springer, 1997
- [4] A. C. Tso, *Gradient Based Learning Methods in Adaptive Processing of Sequences and Data Structures*, Proceedings della "International Summer School on Neural Networks "E.R. Caianiello", Springer, 1997
- [5] J.J. Rajan, *Time series classification*, A thesis submitted to the University of Cambridge for the degree of Doctor of Philosophy, 1994
- [6] D.J.C. Mackay, *Bayesian interpolation*, *Neural Computation*, Vol.4, No 3, 415-447, 1992
- [7] D.J.C. Mackay, *Bayesian Framework for Backpropagation Network*, *Neural Computation*, Vol.4, No 3, 448,472, 1992
- [8] D.J.C. Mackay, *Introduction to Monte Carlo Methods*
- [9] D.J.C. Mackay, *Gaussian Process*
- [10] M.I. Jordan, R.A. Jacobs, *Hierarchical Mixtures of Expert and the EM Algorithm*, MIT 1993
- [11] M.I. Jordan, L. Xu, *Convergence Results for the EM Approach to Mixtures of Experts Architectures*, MIT 1993

- [12] M.I. Jaordan, L. Xu, *On Convergence Properties of the EM Algorithm for Gaussian Mixtures*, MIT 1995
- [13] V. Ramamurti, J. Ghosh, *On the Use of Localized Gating in Mixture of Experts Networks*

Capitolo 4

Ricerca di componenti indipendenti

Un problema fondamentale in molte discipline come la statistica, l'analisi dei dati e il signal processing è trovare una rappresentazione dei dati in modo da metterne in evidenza qualche particolare proprietà.

Recentemente si è sviluppata una tecnica per trovare le componenti indipendenti di una mistura di segnali detta *Independent Component Analysis* (ICA) [3], [4]. Denotiamo con \mathbf{x} una variabile aleatoria m -dimensionale e supponiamo che noi abbiamo una mistura di tali segnali. Sia f una funzione tale che il vettore n -dimensionale $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_n)$ definito da

$$\mathbf{s} = \mathbf{f}(\mathbf{x}) \tag{4.1}$$

ha qualche proprietà desiderata. In molti casi si utilizzano trasformazioni lineari del tipo $\mathbf{s} = \mathbf{W}\mathbf{x}$ ma si possono facilmente estendere i risultati a trasformazioni non lineari [12].

Come indica stesso il nome, l'obiettivo è trovare una trasformazione che rende le componenti s_i statisticamente indipendenti l'una dall'altra. In tal caso il vettore \mathbf{x} rappresenta un mixing non noto di segnali \mathbf{s} generati da sorgenti indipendenti [6], [5].

4.1 Indipendenza statistica

Richiamiamo, in questo paragrafo, alcuni concetti fondamentali. Sia y_1, y_2, \dots, y_m un insieme di variabili aleatorie con media nulla.

Le variabili aleatorie y_1, y_2, \dots, y_m si dicono *statisticamente indipendenti* se la funzione densità di probabilità congiunta $f(y_1, \dots, y_m)$ può essere fattorizzata

$$f(y_1, \dots, y_m) = f_1(y_1)f_2(y_2) \dots f_m(y_m) \quad (4.2)$$

dove $f_i(y_i)$ sono le densità a priori.

Notiamo subito che l'indipendenza deve essere distinta dalla non correlazione in cui

$$E\{y_i y_j\} = E\{y_i\}E\{y_j\} \quad (4.3)$$

L'indipendenza è una richiesta più forte della non correlazione. Infatti se y_i e y_j sono indipendenti si ha

$$E\{g_1(y_i)g_2(y_j)\} = E\{g_1(y_i)\}E\{g_2(y_j)\} \quad (4.4)$$

dove g_1 e g_2 sono due funzioni qualsiasi misurabili.

C'è un solo caso speciale in cui la non correlazione coincide con l'indipendenza statistica: quando le variabili hanno funzione di distribuzione congiunta di tipo Gaussiano. In tal caso la ICA non è applicabile e l'uso della PCA corrisponde alla separazione delle sorgenti.

4.2 Definizione di ICA lineare

Diamo una definizione rigorosa della ICA considerando solo il caso lineare. Nelle definizioni indichiamo con $\mathbf{x} = (x_1, \dots, x_m)^T$ il vettore dei campioni osservati.

La prima definizione, più generale, è

Definizione 1: l'ICA di un vettore aleatorio \mathbf{x} consiste nel trovare la trasformazione lineare $\mathbf{s} = \mathbf{W}\mathbf{x}$ in modo da rendere le componenti s_i più indipendenti possibile nel senso di massimizzare una certa funzione $F(s_1, \dots, s_M)$ di misura di indipendenza statistica.

Questa definizione è troppo generale, non si assume nulla a priori sulla distribuzione dei dati e sulla funzione di misura. Un approccio differente è il seguente:

Definizione 2: il *Noise ICA model* di un vettore \mathbf{x} consiste nello stimare il seguente modello

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \quad (4.5)$$

dove le componenti s_i sono assunte indipendenti. La matrice \mathbf{A} è una matrice di mixing $m \times n$ costante e \mathbf{n} è un vettore di rumore m -dimensionale.

Questo è un problema molto complesso e la maggior parte della ricerca si è concentrata sulla seguente definizione semplificata:

Definizione 3: il *Noise free ICA model* di un vettore \mathbf{x} consiste nello stimare il seguente modello

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (4.6)$$

dove \mathbf{A} ed \mathbf{s} sono definite come per la 4.5.

Il questa tesi ci concentreremo sul modello *noise free* perché sembra che questa semplificazione dia buoni risultati per molti tipi di dati reali.

4.3 Identificabilità di un modello ICA

Le condizioni affinché sia assicurata la separazione tramite la ICA sono le seguenti:

1. Tutte le componenti indipendenti s_i , con la possibile eccezione di una, devono essere non Gaussiane. Infatti, se abbiamo variabili gaussiane la non correlazione implica l'indipendenza e possiamo utilizzare un qualsiasi algoritmo di decorrelazione (per esempio l'analisi di componenti principali)
2. Il numero m di osservazioni devono essere almeno uguale al numero di sorgenti indipendenti n .
3. La matrice A deve avere rango massimo.

Il modello ICA ha anche le seguenti ambiguità:

- Non si possono determinare le varianze delle componenti indipendenti. Infatti, poiché sia s che A sono ignoti, la moltiplicazione di una componente s_j per una costante può essere annullata dividendo la colonna corrispondente a_j di A per la stessa costante. Di conseguenza possiamo, per convenienza matematica, mantenere fisso l'ordine di grandezza delle componenti indipendenti assumendo che abbiano varianza unitaria. Notiamo inoltre che si genera anche una ambiguità nel segno: possiamo moltiplicare la componente indipendente per -1 senza alterare il modello.
- Non possiamo determinare l'ordine delle componenti indipendenti. Questo deriva dalla proprietà commutativa dell'addizione che compare nel prodotto

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{i=1}^n \mathbf{a}_i s_i \quad (4.7)$$

Infatti possiamo esprimere la soluzione come $\mathbf{x} = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}$ dove P è una matrice di permutazione. $\mathbf{P}\mathbf{s}$ sono i vettori indipendenti e $\mathbf{A}\mathbf{P}^{-1}$ è la matrice non nota di mixing.

Come esempio dell'applicazione della ICA per la separazione di componenti indipendenti vediamo in figura 4.1 i tre segnali originali: è stata simulata una sinusoide, un segnale chirp modulato da un esponenziale decrescente e un rumore gaussiano.

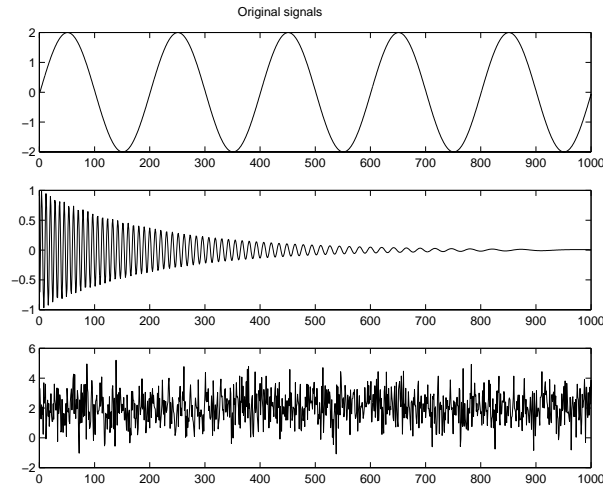


Figura 4.1: Segnali originali

In figura 4.2 possiamo vedere i segnali originali moltiplicati per una matrice di mixing non nota casuale 3×5 , le ricostruzione effettuate dall'algoritmo.

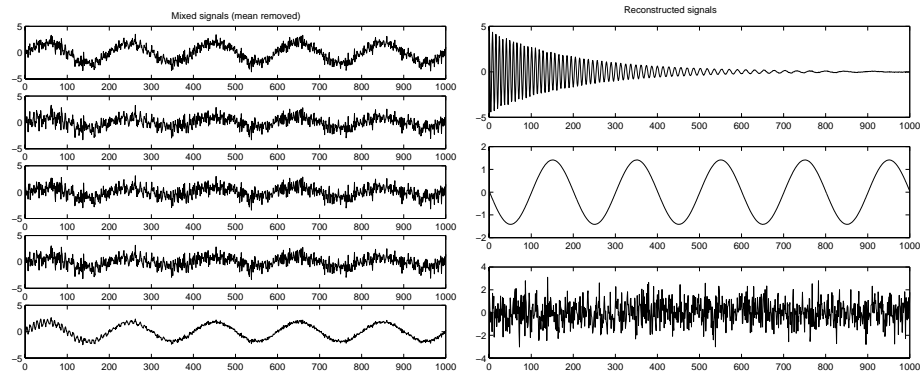


Figura 4.2: Misture e ricostruzione dei segnali

4.4 Funzioni di costo

La stima del modello dei dati delle componenti indipendenti è generalmente ottenuta formulando una funzione di errore, o costo, e poi minimizzandola. In genere si

distingue la funzione di errore dall'algoritmo di minimizzazione, ed il metodo ICA è composto da entrambi separatamente. Anche le proprietà del metodo dipendono da entrambi separatamente ed in particolare:

- le proprietà statistiche, come la robustezza e la varianza asintotica, dipendono dalla funzione di errore
- le proprietà dell'algoritmo, come la velocità di convergenza e la stabilità numerica, dipendono dall'algoritmo di ottimizzazione

Quindi diversi algoritmi di ottimizzazione (ad esempio il gradiente discendente, il gradiente coniugato, i metodi quasi-Newton ecc.) possono essere usati per minimizzare diverse funzioni di errore. In seguito vedremo le principali funzioni di errore ed i principali algoritmi di ottimizzazione utilizzati per stimare le componenti indipendenti.

4.4.1 Verosimiglianza ed entropia della rete

È possibile formulare la verosimiglianza nel modello ICA noise-free in modo da massimizzarla. Denotando con $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_m)^T = \mathbf{A}^{-1}$, il logaritmo della verosimiglianza è

$$L = \sum_{t=1}^T \sum_{i=1}^m \log f_i(\mathbf{w}_i^T \mathbf{x}(t)) + T \ln |\det \mathbf{W}| \quad (4.8)$$

dove f_i sono le funzioni densità di s_i e $\mathbf{x}(t)$, $t = 1, \dots, T$ sono le realizzazioni di \mathbf{x} .

Un'altra funzione di costo deriva dalla teoria delle reti neurali: essa è basata sulla massimizzazione dell'entropia dell'output di una rete con funzioni di attivazioni non lineari [7]. Sia \mathbf{x} l'input della rete e $g_i(\mathbf{w}_i^T \mathbf{x})$ l'output. L'entropia dell'output della rete è

$$L_2 = H(g_1(\mathbf{w}_1^T \mathbf{x}), \dots, g_m(\mathbf{w}_m^T \mathbf{x})) \quad (4.9)$$

Se sono scelte le giuste non linearità g_i , possono essere separate le componenti indipendenti.

Alcuni autori hanno dimostrato che questo approccio, detto Infomax, è equivalente alla stima con la massima verosimiglianza, detta MaxLikelihood, se le non linearità sono le funzioni cumulative di distribuzione delle f_i [2]. Il vantaggio del metodo MaxLikelihood è la stabilità asintotica, gli svantaggi è che bisogna sapere a priori le funzioni di distribuzioni delle componenti indipendenti e che vi è grande sensibilità agli outlier.

4.4.2 Mutua informazione e distanza di Kullback-Leibler

Teoricamente la funzione di errore più valida è la mutua informazione. Definiamo *Entropia Differenziale* di un vettore random \mathbf{y} con funzione di densità di probabilità $\mathbf{f}(\cdot)$ la quantità:

$$H(\mathbf{y}) = - \int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} \quad (4.10)$$

Definiamo *Mutua Informazione* fra m variabili aleatorie $y_i, i = 1, \dots, m$ la quantità:

$$I(y_1, y_2, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{y}) \quad (4.11)$$

Essa rappresenta la misura di indipendenza fra due variabili. La mutua informazione è sempre positiva ed è nulla se e solo se le variabili sono statisticamente indipendenti. Trovare allora una trasformazione che minimizza la mutua informazione, corrisponde a stimare il modello ICA. Per le proprietà della mutua informazione, otteniamo, nel caso di una trasformazione invertibile lineare $\mathbf{y} = \mathbf{W}\mathbf{x}$:

$$I(y_1, y_2, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{x}) - \log |\det \mathbf{W}| \quad (4.12)$$

Il problema principale è che la mutua informazione è difficile da stimare perchè, per utilizzare l'entropia, è necessario calcolare le densità di probabilità.

È possibile tuttavia sviluppare in serie la funzione densità di probabilità $f(\cdot)$, utilizzando i polinomi di Hermite $h(\xi)$ ed i momenti di ordine superiore. Per esempio, per una variabile aleatoria con media nulla e varianza unitaria si ottiene:

$$f(\xi) \approx \phi(\xi)(1 + k_3(y)h_3(\xi)/6 + k_4(y)h_4(\xi)/24 + \dots) \quad (4.13)$$

dove ϕ è la funzione densità di probabilità Gaussiana e $k_i(y)$ sono i momenti i-esimi della variabile y . Possiamo così ottenere un'approssimazione della mutua informazione

$$I(\mathbf{y}) \approx C + \frac{1}{48} \sum_{i=1}^m [4k_3(y_i)^2 + k_4(y_i)^2 + 7k_4(y_i)^4 - 6k_3(y_i)^2 k_4(y_i)^2] \quad (4.14)$$

dove C è una costante; questa approssimazione vale solo se y_i non solo variabili gaussiane.

4.4.3 Tensori di cumulativi di ordine superiore

Un metodo di stima delle componenti indipendenti che è meno direttamente connesso al concetto di funzione obiettivo è legato alla decomposizione nella matrice degli autovettori del tensore dei cumulativi di ordine superiore. Il tensore cumulativo del quarto ordine può essere definito dal seguente operatore lineare T che agisce dallo spazio $m \times m$ in se stesso:

$$T(\mathbf{K})_{ij} = \sum_{k,l} cum(x_i, x_j, x_k, x_l) \mathbf{K}_{kl} \quad (4.15)$$

dove gli indici ij indicano l'elemento (i, j) della matrice \mathbf{K} . Questo è un operatore lineare con m^2 autovalori. Trovando gli autovettori, possono essere stimate le componenti indipendenti.

Il vantaggio principale di questo approccio è che non bisogna conoscere a priori le densità di probabilità delle componenti indipendenti. Poi, come vedremo fra breve, i cumulativi possono essere utilizzati per approssimare la mutua informazione. Lo

svantaggio di questo approccio è che la statistica degli stimatori basati sui cumulativi non sono concordati e poco robusti.

4.5 Algoritmi per la ICA

Dopo aver discusso le possibili funzioni di costo, dobbiamo discutere le possibili implementazioni, cioè, dopo aver scelto la funzione di costo, dobbiamo decidere come ottimizzarla. Le proprietà statistiche del metodo ICA dipendono solo dalla funzione obiettivo usata, il metodo di ottimizzazione influisce sulle proprietà di convergenza. Alcuni algoritmi di ICA necessitano di un preprocessing di whitening dei dati che migliorano le capacità di convergenza.

4.5.1 Algoritmi di decorrelazione non lineare

Diversi algoritmi sono stati introdotti indipendentemente in letteratura per cancellare le correlazioni non lineari fra segnali [4]. Un esempio è l'aggiornamento dei pesi secondo la regola

$$\Delta \mathbf{W} \propto (\mathbf{I} - g_1(\mathbf{y})g_2(\mathbf{y}))\mathbf{W} \quad (4.16)$$

dove $\mathbf{y} = \mathbf{W}\mathbf{x}$ e le non linearità $g_1(\cdot)$ e $g_2(\cdot)$ sono applicate separatamente ad ogni componente del vettore \mathbf{y} .

Un altro algoritmo è stato poi sviluppato recentemente, detto *EASI*, la cui regola di aggiornamento dei coefficienti della matrice di demixing è

$$\Delta \mathbf{W} \propto (\mathbf{I} - \mathbf{y}\mathbf{y}^T - g(\mathbf{y})\mathbf{y}^T + \mathbf{y}g(\mathbf{y}^T))\mathbf{W} \quad (4.17)$$

Il modo migliore per scegliere la non linearità usata in questo algoritmo è utilizzare un approccio di massima verosimiglianza o di massimizzazione dell'informazione, come descritto nella sezione successiva.

4.5.2 Algoritmi per la massima verosimiglianza e stimatori infomax

Una importante classe di algoritmi consiste nella massimizzazione della entropia della rete (infomax) che, sotto alcune condizioni, equivale all'approccio di massima verosimiglianza. Generalmente questo algoritmo è basato sul gradiente discendente stocastico della funzione obiettivo. Per esempio

$$\Delta \mathbf{W} \propto [\mathbf{W}^T]^{-1} - 2 \tanh(\mathbf{W}\mathbf{x})\mathbf{x}^T \quad (4.18)$$

dove la tangente iperbolica è applicata separatamente ad ogni componente del vettore $\mathbf{W}\mathbf{x}$. La tangente iperbolica viene usata perché è la derivata della funzione di distribuzione logistica. Questa funzione di attivazione separa bene segnali con distribuzioni super-gaussiane. Per sorgenti sub-gaussiane bisogna utilizzare un'altra funzione di attivazione.

La convergenza è molto lenta e può essere migliorata utilizzando il gradiente naturale che utilizza la struttura geometrica dello spazio dei parametri [1]. In tal caso la regola di apprendimento è

$$\Delta \mathbf{W} \propto (\mathbf{I} - 2 \tanh(\mathbf{y})\mathbf{y}^T)\mathbf{W} \quad (4.19)$$

con $\mathbf{y} = \mathbf{W}\mathbf{x}$. Con queste modifiche l'algoritmo non necessita di whitening dei dati e converge molto più velocemente.

4.5.3 Algoritmi per la Pca non lineare

Una estensione non lineare della Principal component analysis analizzata nei paragrafi precedenti ci permette di separare non solo componenti gaussiane. Un esempio è la seguente regola di aggiornamento

$$\Delta \mathbf{w}_i \propto g(y_i) - g(y_i) \sum_{j=1}^i g(y_j) \mathbf{w}_j \quad (4.20)$$

dove g è una certa funzione di attivazione. In generale, l'introduzione delle non linearità significa che la regola di apprendimento usa informazioni di ordine superiore, in qualche modo legate a tecniche di ordine superiore (projection pursuit, blind deconvolution e ICA).

Una interessante semplificazione della PCA non lineare è l'algoritmo del bigradiente, con un termine di feedback

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \mu(t)g(\mathbf{W}(t)\mathbf{x}(t))\mathbf{x}(t)^T + \alpha(\mathbf{I} - \mathbf{W}(t)\mathbf{W}(t)^T)\mathbf{W}(t) \quad (4.21)$$

dove $\mu(t)$ è il learning rate dipendente dal passo e α è il momento.

4.5.4 Algoritmo FastIca

Algoritmi adattivi basati sul gradiente discendente stocastico possono essere problematici quando si effettua un'analisi off-line. La convergenza è lenta e dipende drasticamente dalla sequenza del learning rate e dall'inizializzazione dei pesi. Si può usare allora un algoritmo batch basato sull'iterazione a punto fisso, detto *FastIca*, che usa la curtosi per separare le componenti indipendenti [8], [9]. L'algoritmo per una unità e per dati già normalizzati è il seguente:

$$\mathbf{w}(k) = E\{\mathbf{x}g(c(k-1)^T\mathbf{x})\} - E\{g'(\mathbf{w}(k-1)^T\mathbf{x})\}\mathbf{w}(k-1) \quad (4.22)$$

dove il vettore dei pesi \mathbf{w} è normalizzato a norma unitaria ad ogni passo. I valori di aspettazione sono praticamente calcolati ad ogni blocco sufficientemente lungo di dati. Usando più unità è possibile costruire un sistema che stima in parallelo, o in serie più componenti indipendenti.

L'algoritmo FastIca è neurale nel senso che è un sistema di calcolo parallelo e distribuito, ma non adattivo. Invece di usare ogni dato per l'apprendimento, FastIca sfrutta un valore medio computato su un blocco sufficientemente lungo di campioni. Questo aumenta la velocità di convergenza di un fattore 100.

È da notare infine che quando l'algoritmo FastIca è usato con una decorrelazione asimmetrica, è equivalente al metodo di Newton per la stima della massima verosimiglianza. Questo significa che FastIca è un algoritmo generale di ottimizzazione [10].

4.6 Ica con ritardo temporale

Sfortunatamente i mixing istantanei, una delle ipotesi del modello ICA, non avvengono nel mondo reale a causa del filtraggio effettuato dall'ambiente in cui viaggiano i segnali, e dalle differenze dei sensori e dei ritardi di propagazione [11]. Nel mondo reale noi registriamo sempre misture convolute. Per formalizzare il processo di mixing usiamo la *FIR linear algebra* sviluppata da Lambert nel 1996.

Nell'algebra dei filtri FIR gli elementi di matrice sono sostituiti da filtri con risposta impulsiva finita ed è definita la convoluzione elemento per elemento anziché la moltiplicazione standard. Le matrici di filtri sono indicate con una sottolineatura (\underline{A}). Usando questa notazione possiamo esprimere, utilizzando una notazione scalare, il caso di matrice di mixing con convoluzione:

$$x_i(t) = \sum_j \sum_k a_{ij}(k) s_j(t - k) \quad (4.23)$$

o, in forma più elegante

$$\mathbf{x}(t) = \underline{\mathbf{A}} \cdot s(t) \quad (4.24)$$

Utilizzando gli stessi ragionamenti del caso istantaneo possiamo ricavare una regola di apprendimento del tipo

$$\Delta \underline{\mathbf{W}} \propto [\mathbf{I} - f(\underline{\mathbf{W}} \cdot \mathbf{x}(t)) \cdot \mathbf{y}(t)^T] \cdot \underline{\mathbf{W}} \quad (4.25)$$

Diverse variazioni sono state proposte a questo algoritmo, ma non sono stati risolti i problemi di complessità computazionale dovuti all'operazione di convoluzione ed alla presenza di molti minimi locali [14].

È possibile effettuare una trasformata di Fourier della matrice di filtri FIR in modo da ottenere una matrice i cui elementi sono polinomi complessi e la moltiplicazione fra matrici diventa il prodotto fra i polinomi. Indichiamo con $\underline{\hat{A}}$ una matrice polinomiale di filtri FIR. L'equazione di mixing diventa

$$\underline{\hat{X}} = \underline{\hat{A}} \cdot \underline{\hat{S}} \quad (4.26)$$

dove le matrici $\underline{\hat{X}}$ ed $\underline{\hat{S}}$ contengono la trasformata in frequenza rispettivamente di x ed s . Possiamo spezzare l'equazione 4.26 e scrivere

$$\hat{X}_f(t) = \hat{A}_f \cdot \hat{S}_f(t) \quad (4.27)$$

dove $\hat{S}_f(t)$ e $\hat{X}_f(t)$ sono vettori con un elemento per ogni sorgente rappresentante la trasformata in frequenza di s_i e x_i alla frequenza f al tempo t e \hat{A}_f è la matrice contenente gli elementi trasformati dei filtri di mixing alla frequenza f . Abbiamo così ridotto la complessità computazionale trasformando una convoluzione nel dominio temporale in una moltiplicazione nel dominio delle frequenze. Infatti le componenti in frequenza dei sensori sono una mistura istantanea delle componenti in frequenza delle sorgenti. Possiamo così utilizzare un qualsiasi algoritmo sviluppato per la mistura istantanea per separare le registrazioni con convoluzione [15].

Dobbiamo tuttavia effettuare alcune modifiche che ci permettono di lavorare nel piano complesso. Innanzitutto dobbiamo cambiare l'operazione di trasposizione con la coniugazione hermitiana. Con questa modifica l'algoritmo di apprendimento è

$$\Delta \mathbf{W} \propto [\mathbf{I} - f(\mathbf{y}(t)) \cdot \mathbf{y}(t)^H] \cdot \mathbf{W} \quad (4.28)$$

Un'altra importante modifica riguarda la funzione di attivazione. Poichè la tangente iperbolica non è definita in $(k + 1/2)\pi i$ con $k = 0, 1, \dots$, si generano problemi di valutazione numerica e di convergenza. Un approccio alternativo è definire la funzione di attivazione $f(z) = \tanh(\text{Re}\{z\}) + i \tanh(\text{Im}\{z\})$ che non crea problemi di definizione e approssima bene la CDF di una distribuzione gaussiana. Utilizzando

questo algoritmo con le opportune modifiche otteniamo un valore complesso per ogni elemento della matrice di demixing e per ogni intervallo di frequenza ottenuto dalla Short Time Fourier Transform dell'input. Tutti questi elementi formano la risposta in frequenza dei filtri FIR della matrice di demixing.

Con l'approccio nel dominio delle frequenze ci sono però anche alcuni problemi. L'ambiguità di scala si riflette nella differenza delle ampiezze negli intervalli di frequenza, che risulteranno deformati rispetto agli spettri dei segnali originali. Questo problema può essere risolto normalizzando a norma unitaria i coefficienti ad ogni passo dell'algoritmo con

$$\mathbf{W}_f^{norm} = \mathbf{W}_f^{orig} \cdot |\mathbf{W}_f^{orig}|^{-1/N} \quad (4.29)$$

Questa normalizzazione preserva il volume unitario per ogni coefficiente della matrice di demixing e lascia inalterato lo sviluppo spettrale. Per quanto riguarda l'ambiguità per permutazione, non esiste una soluzione soddisfacente anche se un learning rate decrescente, insieme ad un termine di momento, riducono la possibilità di permutazioni random nell'ordine delle componenti indipendenti.

In definitiva l'algoritmo può essere implementato nei seguenti step [13]:

1. Preprocessig dei segnali in modo da filtrare le componenti in frequenza conosciute del rumore e whitening dei dati.
2. Inizializzare nel dominio delle frequenze i filtri di demixing $\underline{\mathbf{W}}$.
3. Prendere un blocco di dati e convertirlo nel dominio delle frequenze con una Short Time Fourier Transform.
4. Filtrare un blocco di input nel dominio delle frequenze $\underline{\mathbf{x}}$ con $\underline{\mathbf{W}}$ per ottenere una stima delle sorgenti $\underline{\mathbf{u}}$.
5. Applicare la funzione di attivazione $f(z) = \tanh(Re\{\cdot\}) + i \tanh(Im\{\cdot\})$.

6. Utilizzare una regola di apprendimento per la variazione $\Delta \underline{\mathbf{W}}$.
7. Prendere il successivo blocco di dati, convertirlo nel dominio delle frequenze e procedere dal punto 4. Ripetere fino alla convergenza.
8. Normalizzare $\underline{\mathbf{W}}$ e convertirlo nel dominio temporale con una IFFT (Inverse Fast Fourier Transform).
9. Ottenere una stima delle sorgenti tramite la convoluzione fra i filtri di demixing $\mathbf{W}(t)$ e le registrazioni $\mathbf{x}(t)$.

Bibliografia relativa al capitolo IV

- [1] S. Amari, A. Cichocki, H. Yang, *A new learning algorithm for blind signal separation*, Advances in Neural Information Processing System 8, MIT Press, Cambridge 1996, pp. 757-763.
- [2] Ahun-Ichi Amari, A. Cichocki, *Adaptive blind signal processing - neural network approaches*, Proceedings of the IEEE, Vol. 86, No. 10, Ottobre 1998
- [3] A. Hyvärinen, *Survey on Independent Component Analysis*
- [4] A. Hyvärinen, E. Oja, *Independent Component Analysis: A Tutorial*
- [5] J. Karhunen, *Neural approach to independent component analysis and sources separation*, Proc. 4th European Symp. on Artificial Neural Networks (ESANN'96), April 1996, pp. 249-266.
- [6] A.J. Bell, T.J. Sejnowski, *An information-maximisation approach to blind separation and blind deconvolution*, Neural Computation, 7:1129-1159, 1995.

- [7] A. Hyvärinen, E. Oja, *Simple neuron models for independent component analysis*, Proc. 1997 IEEE Int. Conf. on Acoustic, Speech, and Signal Processing (ICASSP'97), April 1997, pp. 1483-1492.
- [8] A. Hyvärinen, E. Oja, *A Fixed-Point Algorithm for Independent Component Analysis*, Neural Computation, 1997
- [9] A. Hyvärinen, *Fast and Robust Fixed-Point Algorithm for Independent Component Analysis*, IEEE Trans. on Neural Networks, April 1999.
- [10] X. Giannakopoulos, J. Karhunen, E. Oja, *An Experimental Comparason of Neural Algorithms for Indipendent Component Analysis and Blind Separation*, Internationale Journal on Neural Systems, 9(2), Aprile 1999, pp. 99-122
- [11] Te-Won Lee, A. Bell, R. Orglmeistrer, *Blind Source Separation of Real World Signals*
- [12] A.K. Barros, N Ohnishi, *Improving On-Line Blinde Source Separation of Non-stationaty Signals by Indipendent Component Analysis*, Submitted to IEEE Transactions on Signal Processing.
- [13] A.G. Westner, *Object-Based Audio Capture: Separating Acoustically-Mixed Sounds*, Tesi di "Master of Science in Media Arts and Science", MIT, 1999
- [14] P. Smaragdis, *Blind Separation of Convolutud Mixtures in Frequency Domain*, proceedings of "International Workshop on Indipendence & Artificial Neural Networks"
- [15] A. Bell, T. Sejnowski, *An information-maximization approach to blind source separation and blind deconvolution*, Neural Computation, 7, N0. 6, pp. 1129-1159

Capitolo 5

Esperimenti

5.1 Modellizzazione con un processo NAR

Nell'ambito del progetto *VIRGO* siamo interessati alla rivelazione delle variazioni di temperatura e di rumore acustico. Infatti entrambi sono causa, attraverso l'accoppiamento con l'interferometro, della presenza in uscita di un segnale non generato da una sorgente gravitazionale.

- Poichè le variazioni di temperatura causano delle variazioni della statistica del rumore termico ed un aumento dell'intensità del rumore stesso, le oscillazioni devono limitarsi a $0.1^{\circ}C$ all'interno delle torri (dove sono posti gli specchi, il beam splitter, ecc.). È chiara quindi l'importanza di monitorare tale parametro.
- Il rumore acustico è responsabile del cambiamento della pressione dell'aria con conseguente variazione dell'indice di rifrazione e della posizione degli specchi. Questo causa una anomalia all'uscita dell'interferometro.

Per monitorare la temperatura ed il rumore acustico possiamo considerare la teoria standard della predizione di un segnale con un modello AR (**A**uto **R**egressive).

Un modello AR di ordine P per un segnale discreto $x(n)$ è del tipo:

$$x(n) = a_0 + a_1x(n-1) + \dots + a_Px(n-P). \quad (5.1)$$

Utilizzando varie tecniche ben note nella letteratura, basate sulla risoluzione delle equazioni di Yule-Walker:

$$r_{xx}(k) = \begin{cases} -\sum_{l=1}^P a_l r_{xx}(k-l) & \text{per } k \geq 1 \\ -\sum_{l=1}^P a_l r_{xx}(-l) & \text{per } k = 0 \end{cases} \quad (5.2)$$

dove r_{xx} è la funzione di autocorrelazione, si possono calcolare i coefficienti a_0, \dots, a_P .

Tuttavia se il sistema dinamico che vogliamo modellare con l'equazione 5.1 è non lineare ed è presente del rumore sulle osservazioni dello stesso ordine di grandezza dell'errore minimo di predizione, le performance sono molto scadenti. In tal caso dobbiamo necessariamente utilizzare modelli non lineari, la cui implementazione non si basa purtroppo su un metodo diretto. Le reti neurali, con le loro capacità di generalizzazione, di adattamento e di non linearità, forniscono una soluzione ottimale. I metodi parametrici tradizionali cercano un modello matematico delle caratteristiche del segnale, con il quale effettuare la predizione. Usando una rete neurale possiamo tralasciare la fase di modellizzazione ed effettuare una predizione del segnale su un modello non parametrico non lineare di tipo NAR (**N**on Linear **A**uto **R**egressive).

Denotiamo con $x(n-1), \dots, x(n-n_a)$ i campioni precedenti ad $x(n)$ e concentriamoci sulla predizione del campione successivo, indicata con $\hat{x}(n)$ (one-step-ahead prediction). Il modello NAR corrispondente è descritto dall'equazione:

$$\hat{x}(n) = g(x(n-1), x(n-2), \dots, x(n-n_a)) \quad (5.3)$$

dove $g(\cdot)$ è una funzione non lineare che può essere approssimata da una rete neurale: infatti una rete MLP, con uno strato hidden e funzione di attivazioni non lineari, descrive una combinazione lineare di funzioni non lineari a singola variabile che, in base al teorema di Kolmogorov, può approssimare qualsiasi funzione continua multidimensionale.

In particolare una MLP definisce un modello della dinamica delle variabili misurare. La presenza di anomalie nelle registrazioni di rumore acustico e di temperatura possono essere rivelate analizzando i residui di predizione.

5.1.1 Origine e preprocess dei dati sperimentali

Per poter effettuare una buona stima della dinamica che genera i dati bisogna effettuare una fase di preprocessing. Questa consiste essenzialmente in una stima spettrale, in una conveniente normalizzazione e nella scelta dell'ordine del modello, cioè del numero n_a di ritardi: a questo scopo, nell'ambito della modellizzazione lineare si utilizzano vari stimatori basati sulla varianza dei dati σ , sul numero di dati N e sull'ordine del modello P . I principali sono:

- AIC (Akaike Information Criterium): si ricerca il primo minimo della funzione

$$AIC(P) = N \log \sigma^2(P) + 2P$$

- MDL (Medium Description Length): si ricerca il primo minimo della funzione

$$MDL(P) = N \log \sigma^2(P) + P \log N$$

- FPE (Final Prediction Error): si ricerca il primo minimo della funzione

$$FPE(P) = \sigma^2(P) \frac{N + P + 1}{N - P - 1}$$

Nell'ambito della modellizzazione non lineare dobbiamo conoscere il ritardo ottimale (Time Lag) e la dimensione dello spazio su cui giace la dinamica che si deve identificare (Embedding Dimension). In tal caso possiamo utilizzare gli stimatori utilizzati nell'ambito della teoria del caos, analizzati nel capitolo 2:

- AMI (Average Mutual Information) per il time lag

- FNN (False Nearest Neighbors) per la dimensione di embedding

Il diagramma a blocchi del predittore della serie temporale può essere schematizzato come si vede in figura 5.1

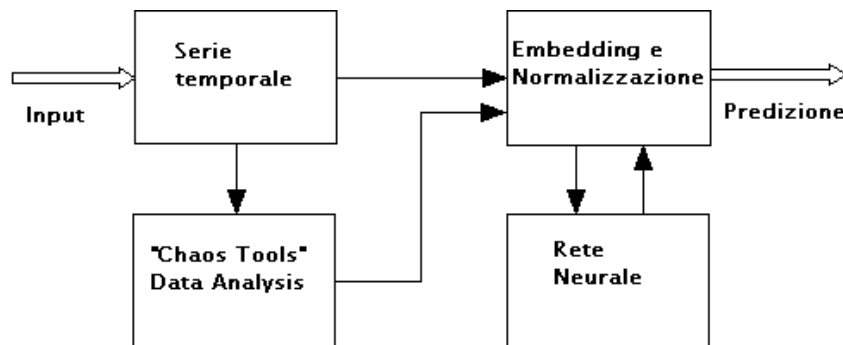


Figura 5.1: Schema a blocchi del predittore di serie temporali

La modellizzazione NAR è stata effettuata su tre serie di dati:

1. la componente x di una serie caotica di Lorenz, mostrata nel capitolo II, per un primo test sulle capacità delle reti.
2. la temperatura registrata a Cascina dal 26 maggio al 7 giugno 1999. I dati consistono in circa 13 giorni di rivelazione continua, con frequenza di campionamento di 0.1 Hz .
3. rumore acustico registrato nel laboratorio *VIRGO*, presso l'Università di Napoli, a Monte Sant'Angelo. La serie consiste in 20 secondi di registrazione con frequenza di campionamento pari a 10000 Hz .

In figura 5.2 e 5.3 si possono vedere gli andamenti temporali dei dati sperimentali ed i relativi spettri ¹.

¹Gli spettri sono stati calcolati utilizzando 2048 campioni, con una finestra di *Hanning* di 1024 campioni ed un *overlap* di 16 campioni

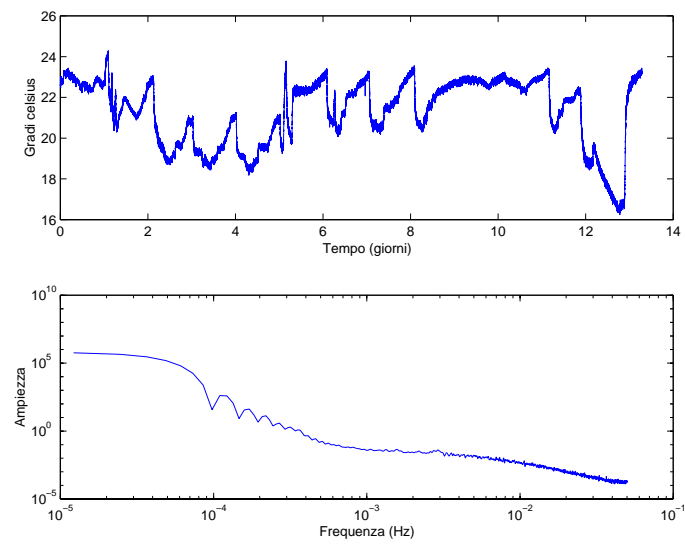


Figura 5.2: Grafico della temperatura e relativa densità spettrale

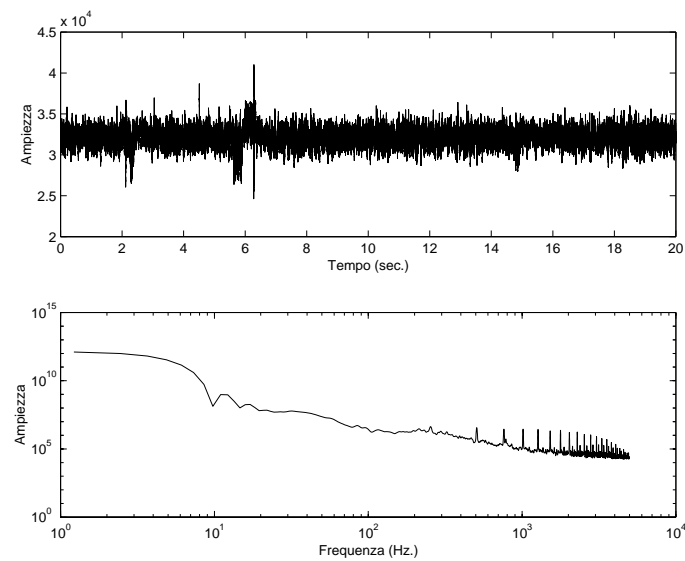


Figura 5.3: Grafico del rumore acustico e relativa densità spettrale

In tabella si possono analizzare i risultati del preprocess per determinare l'ordine del modello.

Criterio	Serie di Lorenz	Temperatura	Rumore acustico
AIC	16	7	26
MDL	11	4	25
FPE	16	7	26
FNN	3	11	20

In base ai risultati noti in letteratura per l'attrattore di Lorenz, la cui dimensione di embedding è pari a 3, abbiamo deciso di utilizzare il False Nearest Neighbours come stimatore più affidabile, anche considerando la natura non lineare dei sistemi in esame. In figura 5.4 si possono vedere i grafici della percentuale di Fnn in funzione della dimensione di embedding per la temperatura e per il rumore acustico.

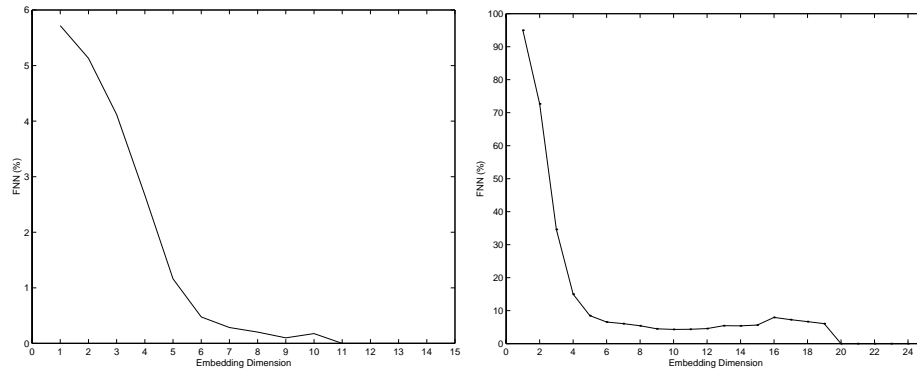


Figura 5.4: Grafici del FNN per la temperatura ed il rumore acustico

5.1.2 Setup del modello NAR

Passo importante per l'implementazione di una rete neurale è la scelta dell'architettura della rete, cioè del numero di neuroni in ogni strato, del numero di strati hidden, delle funzioni di attivazione e del numero di connessioni. Nel capitolo terzo sono

state presentate diverse tipologie di reti per l'identificazione di serie temporali: principalmente o si utilizzano connessioni ricorrenti che memorizzano i valori dei pesi nei vari passi dell'apprendimento (reti FCHLRNN nel caso più generale), o si effettua una codifica spaziale (reti TDNN nel caso più comune).

Abbiamo scelto una rete di Elman con codifica spaziale sull'input ed apprendimento standard per analizzare le performance delle reti ricorrenti ed una rete TDNN con apprendimento Bayesiano per le reti con codifica spaziale pura. Come algoritmo di ottimizzazione abbiamo scelto il gradiente coniugato scalato per entrambe le architetture perché presentava i migliori risultati in termini di velocità ed accuratezza nella convergenza.

Il numero di neuroni di ingresso sono stati scelti pari al numero di ritardi calcolati con il FNN: tre per la serie di Lorenz, undici per la temperatura e venti per il rumore acustico. Come funzione di attivazione è stata scelta la tangente iperbolica. Per lo schema ARD dell'apprendimento Bayesiano sono stati usati un iperparametro per ogni gruppo di connessioni da ogni input allo strato hidden, uno per il gruppo di connessioni dal bias dello strato input allo strato hidden, uno per le connessioni dallo strato hidden all'unità di output ed infine un iperparametro per la connessione dal bias dello strato hidden al neurone di output.

5.1.3 Riepilogo dei risultati

Riportiamo di seguito le tabelle riepilogative dei risultati con le seguenti notazioni:

- npatter: numero di patter utilizzati per l'addestramento
- nepoche: numero di epoche di addestramento
- Nhidden: numero di neuroni dello strato hidden
- Correlazione: coefficiente di correlazione fra il segnale originale ed il segnale generato dalla rete

- Root Mean Square, $rms = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{N}}$
- Normalized Mean Square Error, $NMSE = \frac{\sum_i (x_i - \tilde{x}_i)^2}{N\sigma^2}$
- *Val. Stop.*: sessione di apprendimento interrotta perchè si è raggiunto il minimo dell'errore di validatione
- *msereg*: apprendimento con termine $\propto \sum_i w_i^2$ di regolarizzazione
- *goal to*: sessione di apprendimento interrotta per perché si è raggiunto il minimo di errore prefissato

Serie di Lorenz - Apprendimento Bayesiano						
Nexp	Nhidden	nepoche	npattern	rms	Correlazione	NMSE
exp1	25	100	4000	0.4515	0.979	0.043
exp2	15	500	4000	0.4496	0.980	0.042
exp3	25	500	4000	0.4186	0.985	0.031
exp4	40	100	4000	0.5531	0.953	0.096
exp5	25	1000	4000	0.4051	0.986	0.027
exp6	30	100	6000	0.7588	0.834	0.317
exp7	25	500	6000	0.4285	0.983	0.034
exp8	25	1000	6000	0.4154	0.985	0.030

Rumore acustico - Apprendimento Bayesiano						
Nexp	Nhidden	nepoche	npattern	rms	Correlazione	NMSE
exp1	25	100	10000	0.5316	0.960	0.078
exp2	20	129	50000	1.3618	0.710	1.697
exp3	20	100	20000	0.5384	0.985	0.082
exp4	15	200	10000	0.5254	0.961	0.074
exp5	20	200	20000	0.5365	0.959	0.081
exp6	10	200	10000	0.5285	0.961	0.076

Temperatura - Apprendimento Bayesiano						
Nexp	Nhidden	nepoche	npattern	rms	Correlazione	NMSE
exp1	10	200	10000	0.4076	0.997	0.069
exp2	5	100	20000	0.4910	0.985	0.047
exp3	10	100	9000	0.4814	0.987	0.044
exp4	10	200	9000	0.4763	0.988	0.042
exp5	10	100	18000	0.3165	0.998	0.0071
exp6	10	200	18000	0.3159	0.998	0.0070
exp7	20	200	18000	0.4154	0.994	0.024

Serie di Lorenz - Reti di Elman						
Nexp	Nhidden	nepoche	npattern	rms	Correlazione	NMSE
exp1	15	Val. Stop	4000	0.5552	0.952	0.246
exp2	10	Val. Stop	4000	0.5744	0.946	0.259
exp3	25	Val. Stop	4000	0.5632	0.950	0.264
exp4	40	Val. Stop	4000	0.5636	0.950	0.273
exp5 (msereg)	40	500	6000	0.5710	0.947	0.268
exp6	15	Val. Stop	6000	0.5727	0.946	0.270
exp7 (msereg)	15	500	6000	0.9740	0.550	98.111

Rumore acustico - Reti di Elman					
Nexp	Nhidden	nepoche	npattern	Correlazione	NMSE
exp1	10	Val.Stop-27	10000	0.915	0.191
exp2	15	Val.Stop-16	10000	0.904	0.207
exp3	30	Val.Stop-22	10000	0.912	0.203
exp4	40	Val.Stop-28	10000	0.923	0.183
exp5	25	Val.Stop-36	20000	0.862	0.246
exp6	10	Val.Stop-31	20000	0.868	0.218
exp7	20	Val.Stop-38	20000	0.892	0.203

Temperatura - Reti di Elman						
Nexp	Nhidden	nepoche	npattern	rms	Correlazione	NMSE
exp1	10	goal a 0.01	9000	0.576	0.925	0.234
exp2	10	50	18000	0.546	0.958	0.146
exp3	10	200	18000	0.578	0.956	0.178
exp4	20	100	9000	0.503	0.948	0.193
exp5	20	Val. Stop	18000	0.514	0.957	0.164
exp6	5	goal a 0.01	9000	0.591	0.948	0.191
exp7	5	Val. Stop	18000	0.547	0.954	0.181
exp8 (msereg)	15	200	18000	0.527	0.961	0.121
exp9 (msereg)	25	200	18000	0.581	0.961	0.129
exp10 (msereg)	30	200	18000	0.528	0.962	0.125

Nel grafico 5.5 si può vedere il coefficiente di correlazione e l'errore quadratico medio normalizzato per ogni esperimento effettuato. Si può notare che le migliori performances sono state ottenute, in ogni caso, per le reti TDNN con apprendimento Bayesiano. I motivi principali sono i seguenti:

- Nello schema Bayesiano non è necessario disporre di un validation set perchè la fase di validazione del modello è inclusa nell'apprendimento dei parametri. Nello schema classico è necessario disporre di un data set *indipendente* dal learning set per validare il modello. Questa indipendenza statistica, non è sempre verificabile e può inficiare i risultati, soprattutto per le serie temporali.
- Il problema dell'overfitting non è presente nello schema Bayesiano perchè i pesi sono automaticamente regolarizzati. Nello schema classico all'apprendimento l'overfitting è un'inconveniente di difficile soluzione.
- Dalla stima degli iperparametri possiamo effettuare direttamente una stima

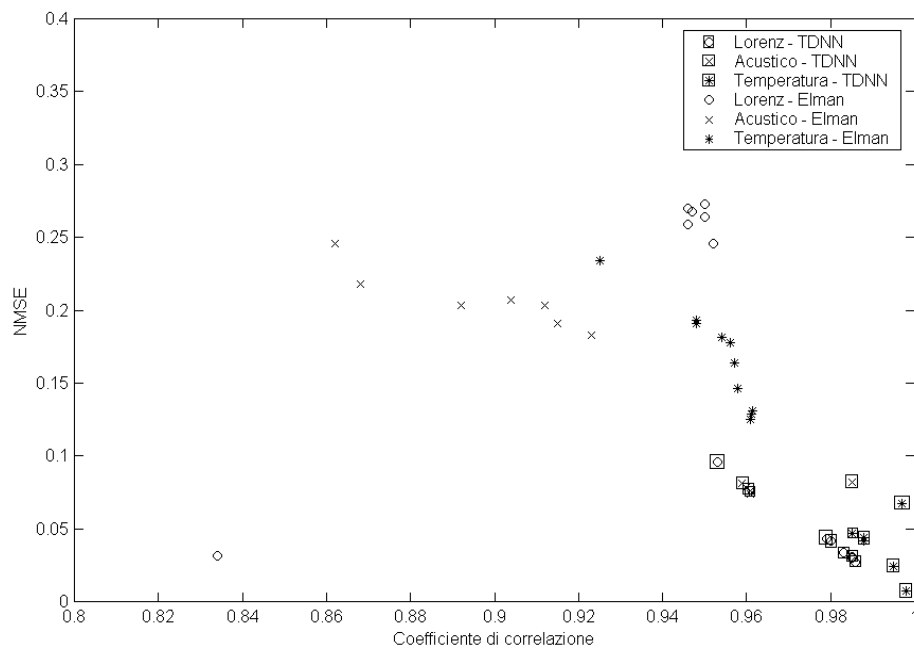


Figura 5.5: Performance dei modelli predittivi

della significatività degli input e delle connessioni tra gli strati. Questa analisi non è praticabile nello schema classico.

- La struttura delle reti ricorrenti può generare stati di equilibrio spuri, come i cicli limite, che non rappresentano l'ottimo dell'apprendimento.
- Dal punto di vista computazionale le tecniche Bayesiane sono più economiche.

Presentiamo ora i grafici dei risultati migliori.

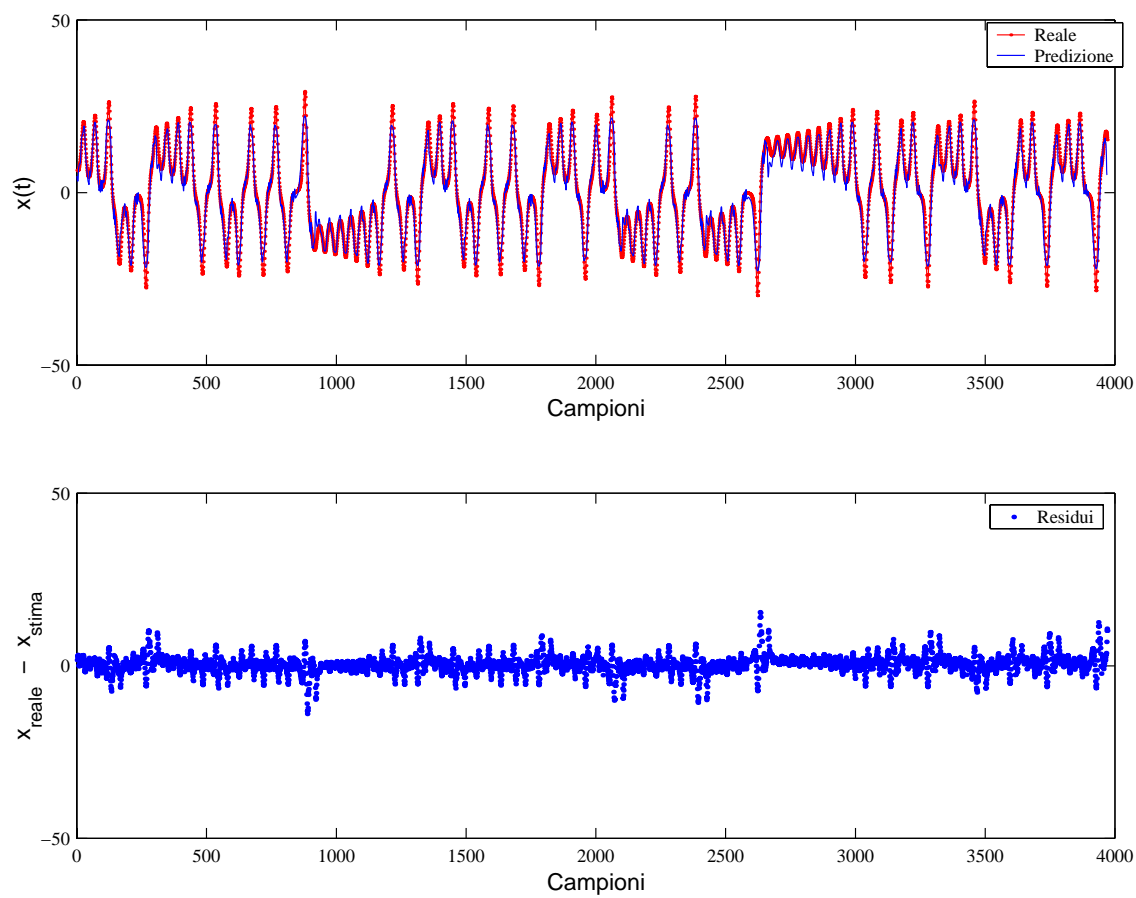


Figura 5.6: Predizione per la serie di Lorenz

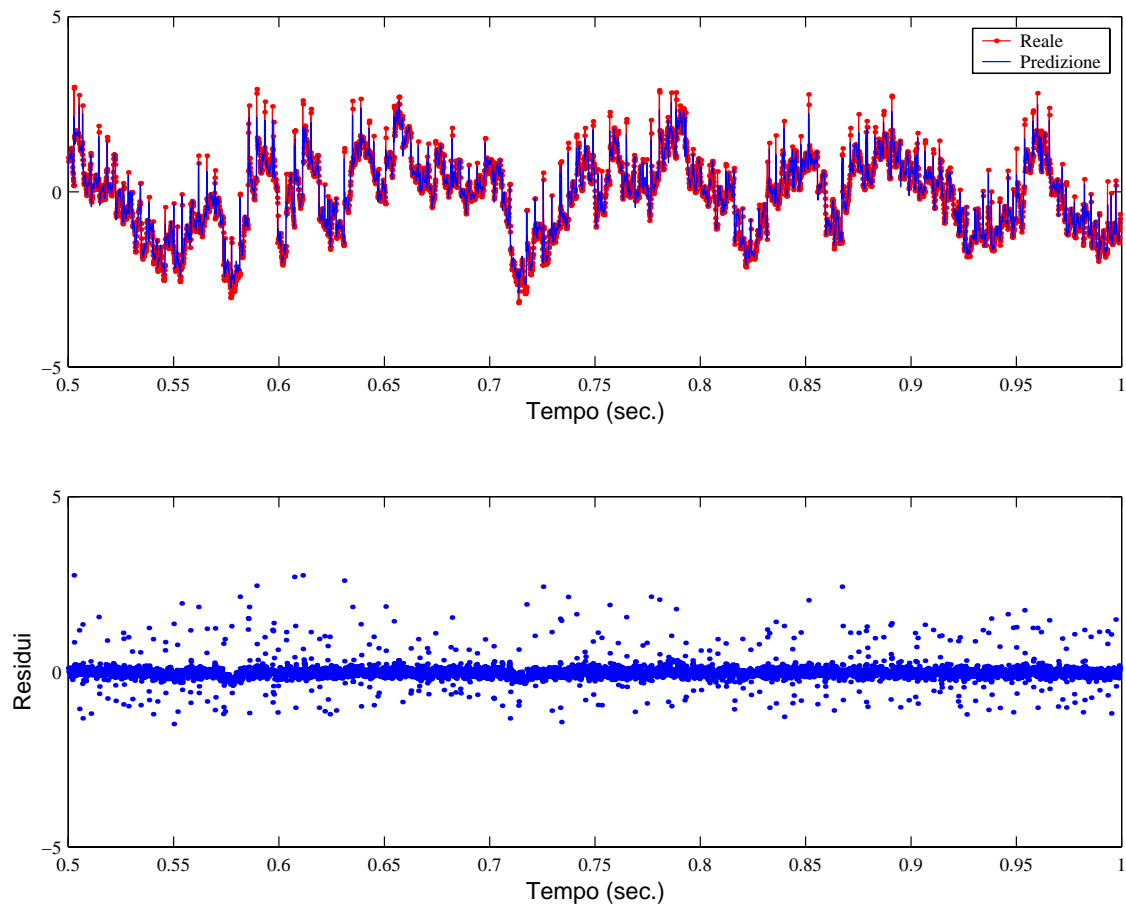


Figura 5.7: Predizione per il rumore acustico

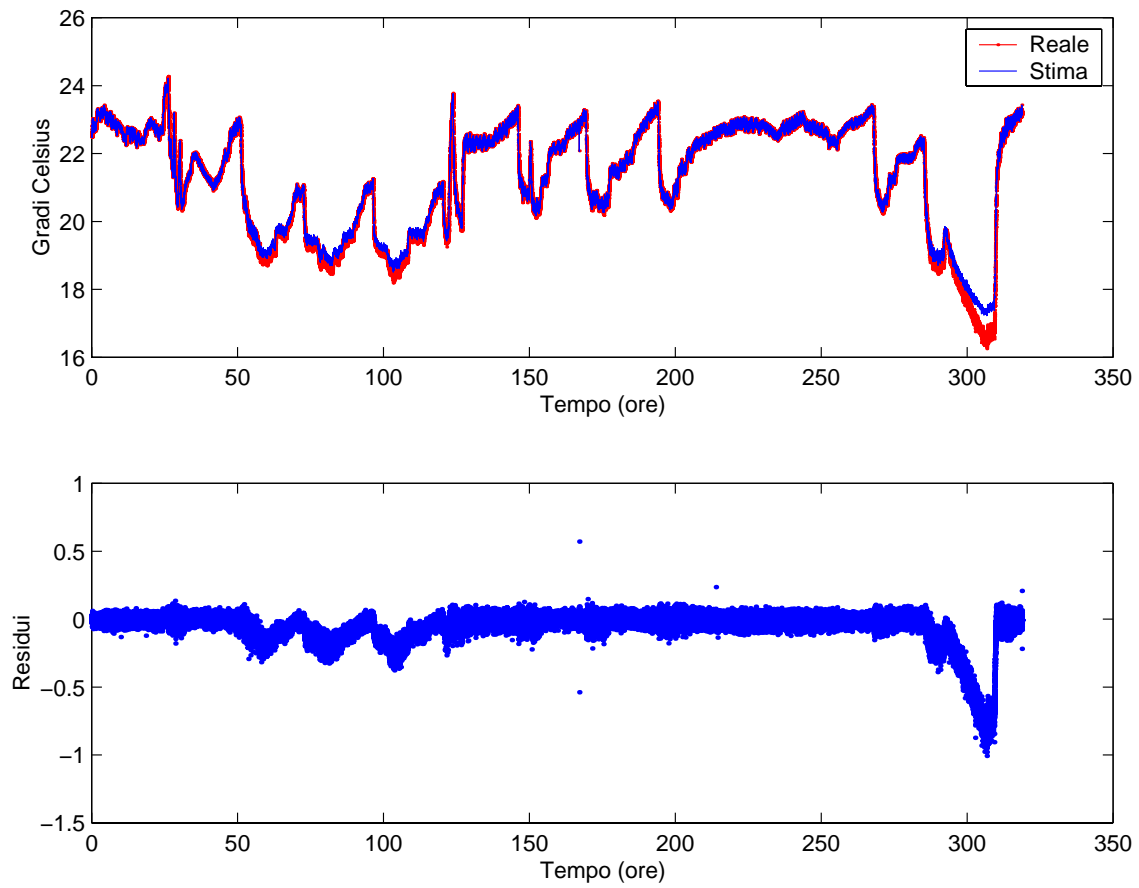


Figura 5.8: Predizione per la temperatura

Come si può vedere dai grafici, il problema è molto complesso. Tuttavia, dall'analisi dei residui, si nota che le reti TDNN con apprendimento Bayesiano hanno mostrato una buona capacità di predizione. Le reti possono predire anomalie la cui varianza non supera la varianza dei residui. Se fissiamo a due o tre σ dei residui la soglia per la rivelazione, possiamo effettuare un trigger per selezionare frames di dati da indirizzare verso un'analisi più raffinata ma dispendiosa.

In conclusione, un modello NAR per il rumore ambientale può essere implementato non solo per predire anomalie che possono correlarsi con l'uscita dell'interferometro e compromettere la rivelazione, ma anche per ridurre la quantità di dati da analizzare, soprattutto in funzione dell'oneroso costo computazionale del matched filter.

5.2 Modellizzazione NARMA per l'esperimento *VIRGO*

La conoscenza delle caratteristiche statistiche della distribuzione di rumore dell'apparato è essenziale per la ricerca dei segnali gravitazionali per due motivi principali:

- dobbiamo conoscere accuratamente lo spettro per effettuare l'operazione di *whitening* del rumore.
- abbiamo necessità di simulare i dati di rumore nel tempo per verificare le prestazioni del matched filter.

Il problema principale è: dato lo spettro $S(f)$, trovare il corrispondente processo stocastico discreto nel dominio del tempo $x(n)$.

Il problema dell'identificazione del processo stocastico dalla conoscenza dello spettro può essere riformulato nel problema dell'identificazione non lineare del sistema nel seguente modo:

1. noto lo spettro, si genera la corrispondente serie nel dominio del tempo
2. si genera un ingresso $u(n)$, tipicamente rumore gaussiano
3. si effettua la convoluzione fra l'ingresso e la serie temporale del sistema per ricavare l'uscita $y(n)$
4. utilizzando la coppia ingresso-uscita, si ricerca un modello non lineare nella forma:

$$y(n) = F(y(n-1), y(n-1), \dots, y(n-n_y), u(n), u(n-1), \dots, u(n-n_u)) \quad (5.4)$$

Nell'ambito della teoria lineare, in cui $F(\cdot)$ si riduce ad una costante, il modello 5.4 è detto ARMA² ed i coefficienti possono ricavarsi risolvendo le equazioni di Yule-Walker per modelli ARMA, simili alle 5.2. Per effettuare una identificazione non lineare, utilizziamo le reti neurali per le proprietà già citate precedentemente. Il modello per l'identificazione del sistema è mostrato schematicamente in figura 5.9

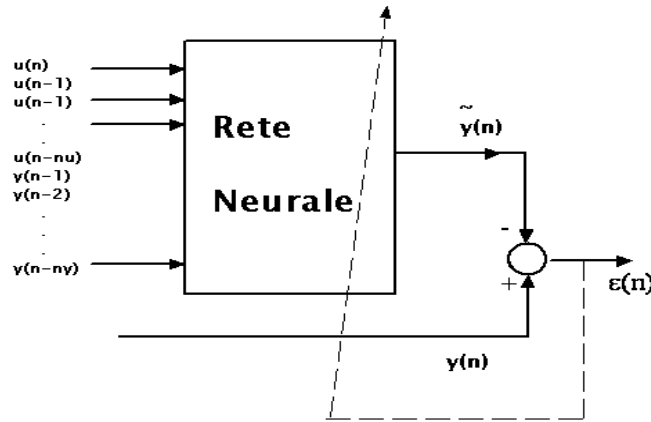


Figura 5.9: Schema per l'identificazione di un sistema

²Auto **R**egressive **M**oving **A**verage

Nelle simulazioni numeriche abbiamo utilizzato la seguente parametrizzazione "one side" per il rumore di *VIRGO* [3]:

$$S(f) = \begin{cases} \frac{S_1}{f^5} + \frac{S_2}{f} + S_3 \left(1 + \left(\frac{f}{f_K} \right)^2 \right) + S_v(f) & \text{per } f > 0 \\ 0 & \text{altrimenti} \end{cases} \quad (5.5)$$

dove:

$$\begin{aligned} f_K &= 500 \text{ Hz} && \text{Frequenza di taglio per il rumore shot} \\ S_1 &= 1.08 \cdot 10^{-36} && \text{Pendulum noise} \\ S_2 &= 0.33 \cdot 10^{-42} && \text{Mirror noise} \\ S_3 &= 3.24 \cdot 10^{-46} && \text{Shot noise} \end{aligned} \quad (5.6)$$

Il contributo dei modi di violino dei fili (Violin noise) è stato parametrizzato come:

$$S_v(f) = \sum_n \frac{1}{n^4} \frac{f_1^{(c)}}{f} \frac{C_c \phi_n^2}{\left(\frac{f^2}{f_n^{(c)2}} - 1 \right)^2 + \phi_n^2} + (c \leftrightarrow f) \quad (5.7)$$

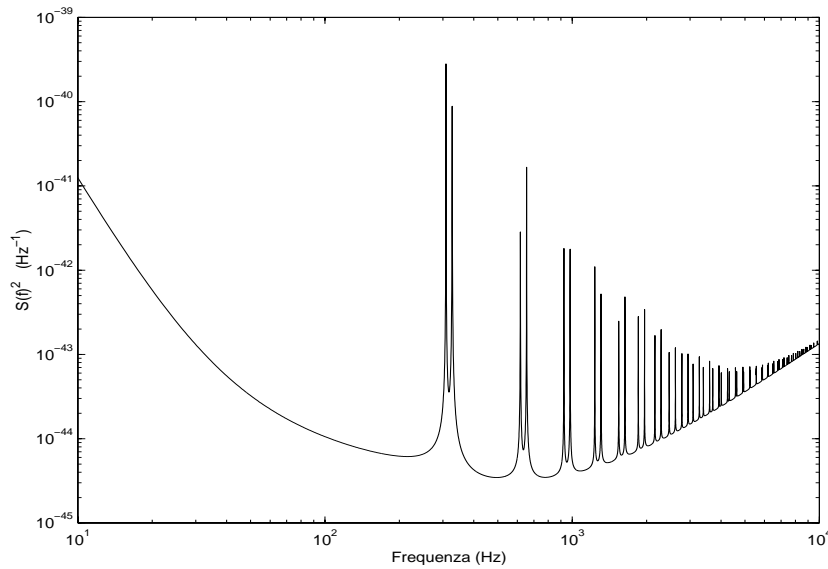
dove abbiamo tenuto conto del fatto che gli specchi vicini e quelli lontani hanno masse differenti. Le costanti sono:

$$\begin{aligned} f_n^{(c)} &= n \cdot 327 \text{ Hz} & f_n^{(f)} &= n \cdot 308.6 \text{ Hz} \\ C_c &= 3.22 \cdot 10^{-40} & C_f &= 2.82 \cdot 10^{-40} & \phi_n^2 &= 10^{-7} \end{aligned}$$

In figure 5.10 si può vedere il modello teorico semplificato dello spettro nell'intervallo 10-10000 Hz.

5.2.1 Setup del modello NARMA

In considerazione dei risultati ottenuti per la modellizzazione NAR, abbiamo utilizzato come architettura di base una rete TDNN con apprendimento Bayesian.

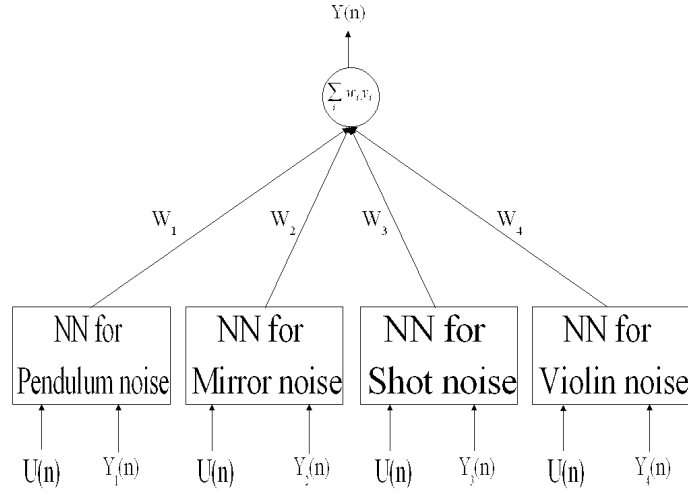
Figura 5.10: Spettro modellato di *VIRGO*

Da precedenti lavori [16], sappiamo che una singola rete non riesce ad identificare tutte le caratteristiche dello spettro: in particolare le risonanze ad alta frequenza presentano una variazione in potenza di 3-4 ordini di grandezza che una singola rete non riesce a modellare.

Per superare questo problema abbiamo usato una rete per ogni sorgente di rumore, addestrandola con la serie temporale il cui spettro è una componente dello spettro totale. Poichè il rumore totale è una combinazione lineare di ogni singola sorgente, possiamo usare una ADALINE³ per effettuare una mistura degli output di ogni singola rete. Nei nostri esperimenti abbiamo usato 4 MLP, una per ogni sorgente di rumore: shot noise, mirror noise, pendulum noise e violin noise. Lo schema è quello di figura 5.11.

Il numero di neuroni input è pari alla somma del numero di ritardi sull'uscita più il numero di ritardi sull'ingresso. Il numero di neuroni hidden ed il numero di pattern di addestramento sono variabili. Tutte le funzioni di attivazione sono

³Una ADALINE è una rete lineare che esegue i minimi quadrati in modo adattivo

Figura 5.11: Spettro modellato di *VIRGO*

tangenti iperboliche per lo strato hidden ed i neuroni di output sono lineari.

5.2.2 Esperimenti con $F_{max} = 1024 \text{ Hz}$

La prima serie di esperimenti è consistito nella modellizzazione dello spettro fra 10 e 1024 Hz. Il numero di neuroni di ingresso sono stati scelti in base alla significatività dell'iperparametro corrispondente ed al criterio del FNN: abbiamo scelto il numero di ritardi pari a cinque per la parte a media mobile ed 8 per la parte autoregressiva. I pattern di input e di target sono stati normalizzati a media zero e varianza unitaria. Come parametro di precisione abbiamo utilizzato l'iperparametro β dell'apprendimento Bayesiano che è proporzionale all'inverso della varianza dei residui di predizione; l'inizializzazione pone $\beta = 50$ e quanto più il valore cresce tanto migliora il modello. Riportiamo di seguito la tabella riepilogativa dei risultati.

Spettro fino a 1024 Hz				
Nexp	Nhidden	nepoche	Secondi	β
exp1	5-5-5-5	100	10	471-709-819-115
exp2	5-5-5-15	100	20	591-736-912-129
exp3	5-5-5-5	200	20	512-762-964-124
exp4	5-5-5-5	100	30	589-786-970-120
exp5	10-10-10-10	100	30	979-925-606-135
exp6	10-10-10-10	200	30	985-954-624-145

Gli output delle reti nell'esperimento *exp6* sono stati utilizzati per effettuare la mistura con la rete ADALINE. Il coefficiente di correlazione fra la serie generata dallo spettro e la serie generata dal modello è 0.9991 e l'errore quadratico medio è 0.00119.

Come si può notare da questi primi esperimenti il modello è abbastanza buono. La varianza dei residui si mantiene intorno al 5% della varianza del segnale. La potenza spettrale dei residui è 4 ordini di grandezza rispetto alla potenza dell'uscita predetta. Ovviamente le performance possono essere migliorate aumentando la complessità della rete, cioè il numero di neuroni hidden, e la precisione nella convergenza, aumentando il numero di epoche di apprendimento⁴. Tuttavia, benché i residui seguino abbastanza bene una distribuzione gaussiana, come richiesto dal modello di distribuzione che abbiamo ipotizzato, non si ottiene una buona operazione di whitening dello spettro.

5.2.3 Introduzione del rumore di *creep*

Come notato nel capitolo II, nello spettro di rumore dell'esperimento *VIRGO* sono presenti due risonanze a bassa frequenza, rispettivamente a circa 6.5 e 6.8 Hz. Queste

⁴Ricordiamo che le reti con apprendimento Bayesiano non risentono degli effetti di overfitting

risonanze, chiamate *creep*, sono dovute ai cedimenti microscopici a cui sono soggetti i fili di sospensione. Passo successivo del lavoro di tesi è l'introduzione di questa ulteriore sorgente di rumore e la successiva modellizzazione.

In analogia con le risonanze di violino, le risonanze di creep sono state modellate come:

$$S_c(f) = \frac{C_r^2 \xi^2}{\left[\left(\frac{f}{f_r}\right) - 1\right]^2 + \xi^2} + \frac{C_q^2 \xi^2}{\left[\left(\frac{f}{f_q}\right) - 1\right]^2 + \xi^2} \quad (5.8)$$

dove le costanti sono:

$$f_r = 6.5 \text{ Hz} \quad f_q = 6.8 \text{ Hz}$$

$$C_r^2 = 2 \cdot 10^{-19} \quad C_q^2 = 1 \cdot 10^{-19} \quad \xi^2 = 1 \cdot 10^{-4}$$

In figura 5.2.3 si vede lo spettro fra 1 e 5000 Hz.

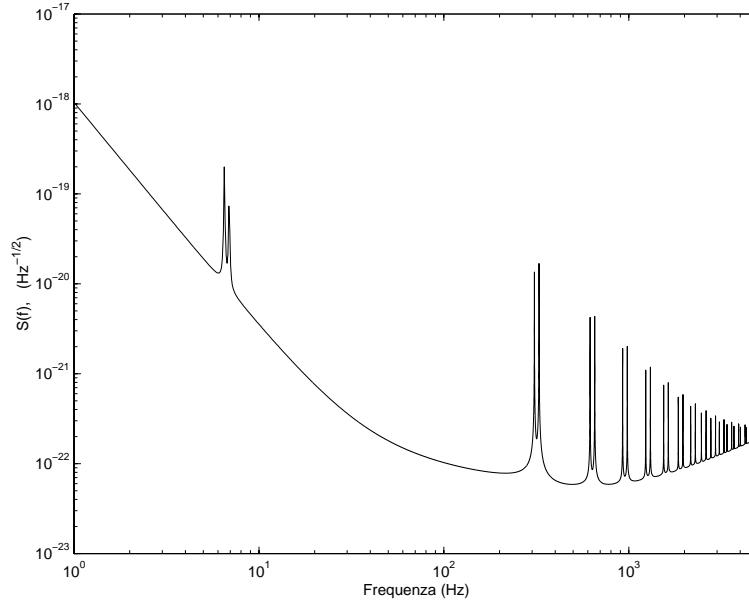


Figura 5.12: Spettro di rumore di *VIRGO* incluso il creep noise

In questa nuova serie di esperimenti la frequenza minima dello spettro è 1 Hz e la frequenza di campionamento di 10000 Hz ci permette di simulare fino a 5000 Hz. Abbiamo introdotto una quinta rete per modellare il rumore di creep ed abbiamo

usato il metodo dei minimi quadrati, con una rete ADALINE, per combinare i risultati delle cinque reti.

I neuroni di ingresso sono 10 per ogni rete: 5 per la parte MA e cinque per la parte AR. Il numero di neuroni hidden è variabile per ogni esperimento e per ogni rete. Tutte le funzioni di attivazione sono tangenti iperboliche e l'addestramento dura 100 epoche. La tabella dei risultati è la seguente:

Spettro fino a 2048 Hz con creep noise					
Nexp	Secondi	nhidden	β	NMSE	Correlazione
exp1	5	4-4-4-8-6	602-237-744-84- $1.3 \cdot 10^4$	0.546	0.985
exp2	30	8-8-10-12-10	$1.3 \cdot 10^3$ -87-64-19- $5.05 \cdot 10^3$	0.287	0.994
exp3	60	10-10-10-15	$1.5 \cdot 10^3$ -24.4-15.5-16.9-631	0.234	0.998

Anche in questo caso i residui si distribuiscono abbastanza bene secondo una funzione di distribuzione gaussiana. Tuttavia non si riesce ad ottenere un buon *whiten-ing* dello spettro su tutto l'intervallo di frequenze. Questo è dovuto alla complessità dei modi di creep e dal basso numero di secondi di dati utilizzati nell'addestramento.

5.2.4 Riduzione dello spettro a 1000 Hz

Scopo principale di questo esperimento è capire se la rete riesce ad rendere bianco lo spettro dei residui ed identificare le risonanze. In conseguenza abbiamo pensato di riparametrizzare lo spettro in modo da mantenere la stessa forma e, riducendo la frequenza massima di identificazione, utilizzare un patter temporale più lungo. Campionando a 1000 Hz, lo spettro è ridotto all'intervallo 1 – 500 Hz e parametrizzato riducendo la frequenza di rumore shot a $f_K = 50$ Hz, anticipando le frequenze di risonanza dei modi di violino di un fattore 30 e riducendo la costante del rumore f^{-5} (rumore di pendolo) di cinque ordini di grandezza. In questo modo i modi di

violino sono lontani dalla frequenza di Nyquist e non risentono dei problemi del campionamento.

Le reti conservano la struttura considerata negli esperimenti precedenti. Abbiamo una MLP addestrata con apprendimento Bayesiano per ogni sorgente di rumore. Il numero di neuroni di ingresso sono 8 per la parte AR e 5 per la parte MA. I neuroni hidden, tutti con funzione di attivazione tangente iperbolica, sono 15 per le reti che modellano le risonanze e 10 per le altre. Il neurone dell'ultimo strato è lineare. In figura 5.2.4 possiamo vedere lo spettro dei residui. Come si può notare, l'operazione di *whitening* è avvenuta nel range di frequenze $70 - 500 \text{ Hz}$. Lo spettro dei residui

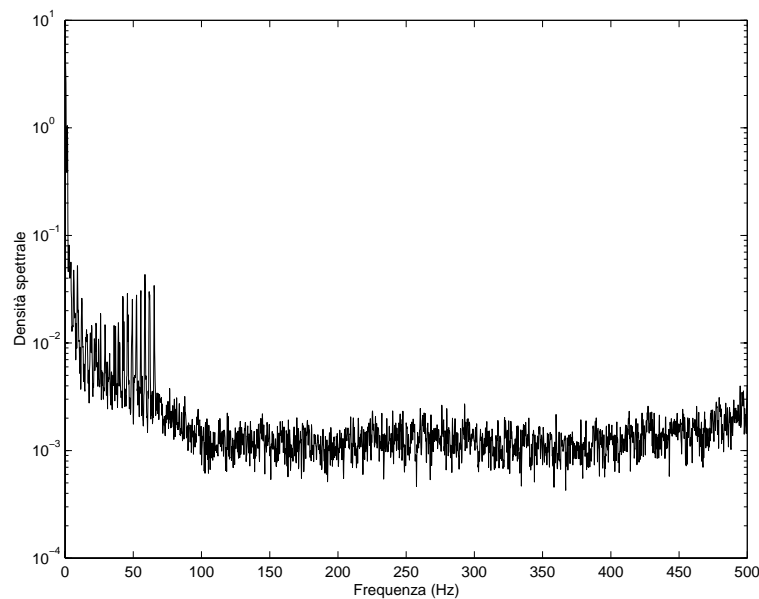


Figura 5.13: Spettro dei residui

conserva la struttura della curva di sensibilità a basse frequenze a causa della limitatezza del patter di addestramento. Infatti, aumentando il numero di campioni, la frequenza minima per cui lo spettro dei residui è bianco diminuisce; al contrario, diminuendo il numero di campioni, la frequenza massima per lo spettro è bianco aumenta.

5.2.5 Esperimenti con $F_{max} = 2048 Hz$

Nei successivi esperimenti abbiamo campionato i dati a $F_c = 4096 Hz$, abbiamo considerato altre risonanze dei modi di violino e considerato la banda frequenza $10 - 2048 Hz$. Lo schema di mistura con quattro reti TDNN con apprendimento Bayesiano (una per ogni sorgente di rumore) è uguale agli esperimenti precedenti. Le reti sono state addestrate con 13 neuroni di ingresso (7 per la parte a media mobile e 6 per la parte autoregressiva) e sono a singolo strato hidden con 5 neuroni con funzione di attivazione tangente iperbolica. Solo nell'ultimo esperimento (exp5) la quarta rete ha 15 neuroni hidden, la dimensione della parte autoregressiva è variabile ⁵ e le reti sono state addestrate per 200 epoche. Il numero di patterns utilizzate nell'apprendimento sono stati aumentati progressivamente per capire che punto avveniva l'operazione di whitening. Presentiamo la tabella riepilogativa degli esperimenti. I valori dell'errore quadratico medio normalizzato (NMSE) e del coefficiente di correlazione sono riferiti alle serie temporale teorica ed a quella generata dal modello di mistura delle reti.

Spettro fino a 2048 Hz				
Nexp	Secondi	β	NMSE	Correlazione
exp1	5	551-784-964-476	0.014	0.999
exp2	10	498-516-944-319	0.062	0.998
exp3	20	261-607-774-323	0.094	0.998
exp4	30	423-334-775-256	0.029	0.998
exp5	30	581-1710-1284-728	0.011	0.9997

Nonostante il modello generato dalle reti presenti un buon accordo con la curva di sensibilità (figura 5.14) ed i residui si distribuiscono in secondo una gaussiana (figura 5.15, come richiesto dal modello di distribuzione che abbiamo scelto per l'ap-

⁵Dall'analisi del FNN per l'uscita si ottiene 7 per la prima rete, 6 per la seconda, 4 per la terza e 8 per la quarta

prendimento Bayesiano, l'operazione di *whitening* non è avvenuta su tutto l'intervallo di frequenze fino 30s di training. Questo è dovuto ai lunghi tempi di correlazione fra

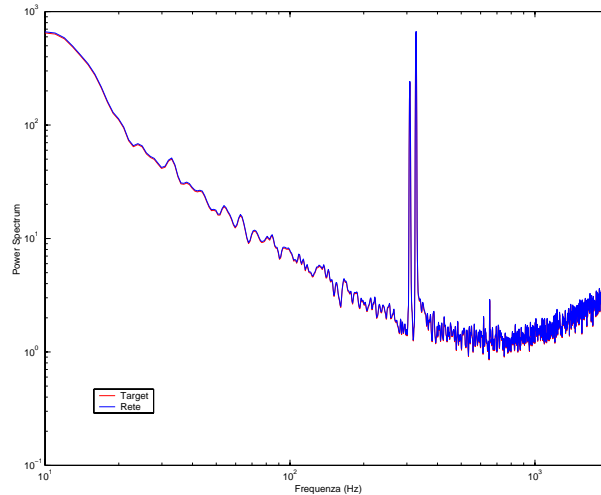


Figura 5.14: Spettro fino a 2048 Hz dell'uscita teorica e dell'uscita modellata dalla rete per l'esperimento 5

ingresso uscita: infatti, se il numero di secondi utilizzati nella fase di training non superano i tempi di correlazione fra ingresso ed uscita della funzione di trasferimento dell'interferometro, la dinamica a basse frequenze, dove l'escursione in ampiezza è maggiore, non può essere identificata ed i residui seguono lo stesso andamento della curva di sensibilità.

Alla luce di queste considerazioni abbiamo effettuato un ultimo esperimento utilizzando per l'apprendimento 240 secondi di dati. L'analisi degli iperparametri relativi ai neuroni di ingresso ci ha indicato la dimensione dell'input: 5 per la parte MA e 8 per la parte AR. Lo strato hidden è composto da 8 neuroni con funzione di attivazione tangente iperbolica per le prime tre reti e 20 per la rete che modella i modi di violino. Le reti sono state addestrate per 200 epoche con apprendimento bayesiano e come algoritmo di ottimizzazione è stato utilizzato il gradiente coniugato scalato. I risultati sono stati un coefficiente di correlazione di 0.9812 e un errore

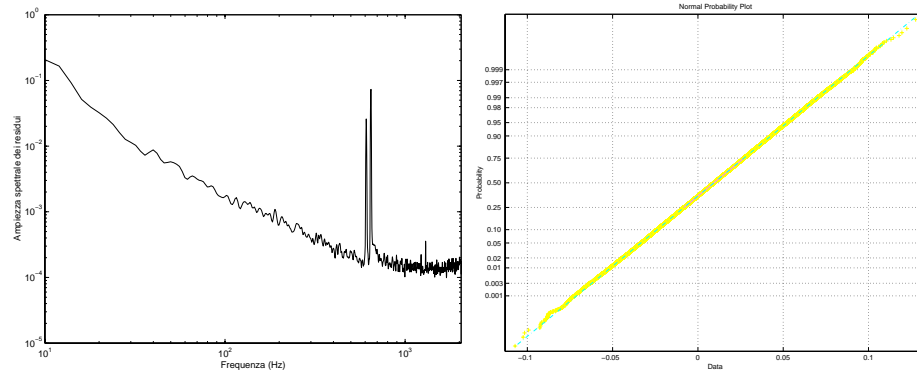


Figura 5.15: Spettro ed normplot dei residui per l'esperimento 5

quadratico medio normalizzato di 0.45. In figura 5.16 si possono vedere i risultati dell'esperimento: lo spettro dei residui ed il normplot mostrano che i residui seguono una distribuzione gaussiana. L'operazione di whitening è avvenuta, considerando che la deviazione ad alte frequenze è generata da effetti di campionamento dovuti alle oscillazioni che si generano vicino alla frequenza di Nyquist.

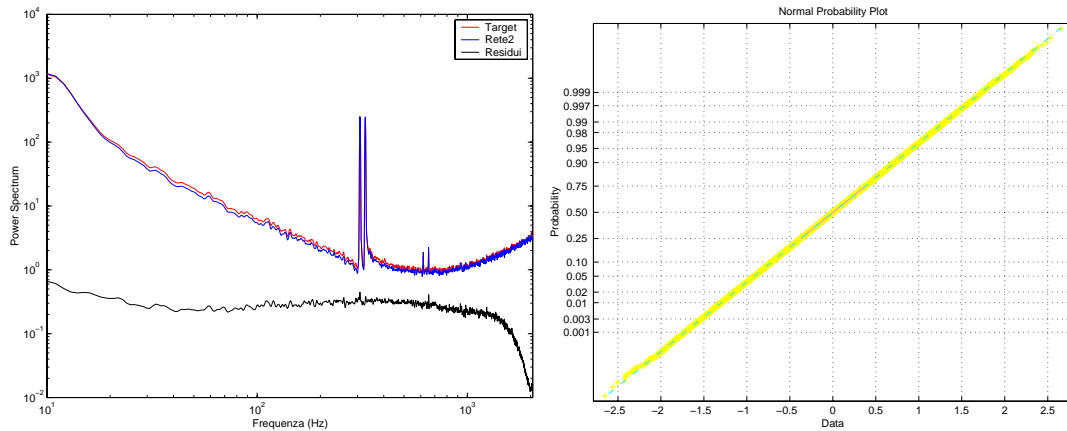


Figura 5.16: Spettro del segnale originale, predetto e dei residui (sx) e normplot dei residui

5.2.6 Matched filter per la rivelazione del segnale

Una volta identificato il processo stocastico che genera il rumore dell'interferometro, abbiamo il filtro di whitening per applicare il Matched filter. Come forma d'onda in ingresso abbiamo scelto un chirp nella parametrizzazione analizzata da Marano:

$$s(t) = IA(t; t_0, \tau) \cos[\beta(t; t_0, \tau) + phi] \quad (5.9)$$

con

$$A(t; t_0, \tau) = \frac{1}{\tau f_0^2} \left(1 - \frac{t - t_0}{\tau} \right)^{-1/4}$$

$$\beta(t; t_0, \tau) = \frac{16\pi f_0 \tau}{5} \left(1 - \left(1 - \frac{t - t_0}{\tau} \right)^{5/8} \right) \quad (5.10)$$

$$(5.11)$$

dove t_0 è l'istante in cui inizia il segnale, f_0 è la frequenza iniziale, $\tau = (M_{rid}/M_\odot)$ è la durata del segnale ed I è una costante di normalizzazione che rende l'energia del chirp unitaria. La tecnica di rivelazione si basa sul calcolo del rapporto segnale-rumore massimo che si può ottenere, dato un certo *template*. Scelta una soglia di rivelazione come percentuale del SNR massimo (per esempio il 90% o 95%) , si valuta l'uscita del matched filter. Quando l'uscita del matched filter supera la soglia possiamo segnalare la rivelazione. Se più di un template è indicato come segnale possibile, si valuta il rapporto segnale-rumore maggiore.

In figura 5.2.6 è presentato un possibile segnale gravitazionale generato da binarie coalescenti con parametri $I = 13077$, $t_0 = 14$, $f_0 = 100$, $M/M_\odot = 1.71$ e la sua registrazione corrotta dal rumore dell'interferometro. Come modello della curva di sensibilità abbiamo scelto quello che presentava lo spettro dei residui più bianco, cioè l'esperimento con frequenza di campionamento $F_c = 4096 \text{ Hz}$ e addestramento con 240 s di dati. Come soglia abbiamo scelta il 90% del rapporto segnale-rumore massimo o in modo equivalente il 90% della correlazione.

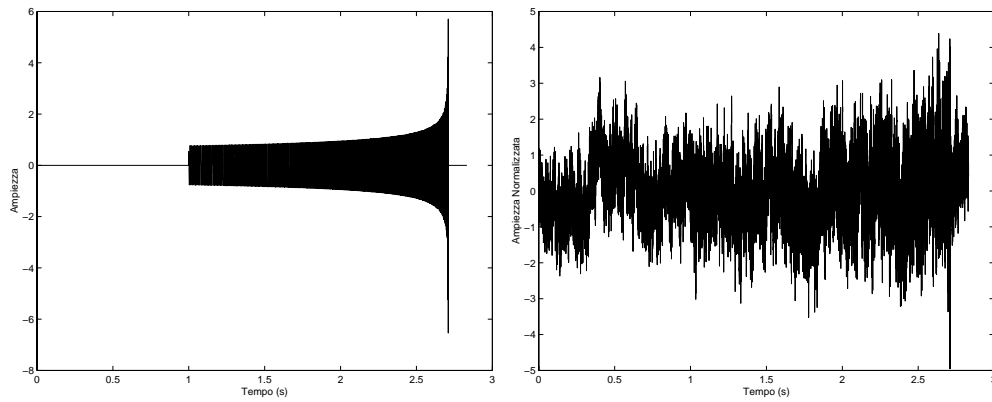


Figura 5.17: Segnale in ingresso all'interferometro e segnale corrotto dal rumore

In figura 5.2.6 possiamo vedere la correlazione fra il template ed i residui di predizione del modello identificato dalla rete. Nel grafico a sinistra il segnale è presente nella registrazione ed è stato correttamente rivelato; nel grafico a destra è presente solo il rumore e correttamente non è stato rivelato perchè la correlazione non ha superato la soglia. Si può anche notare che il massimo della correlazione identifica correttamente l'istante di tempo in cui arriva il segnale ($t = 1$ s).

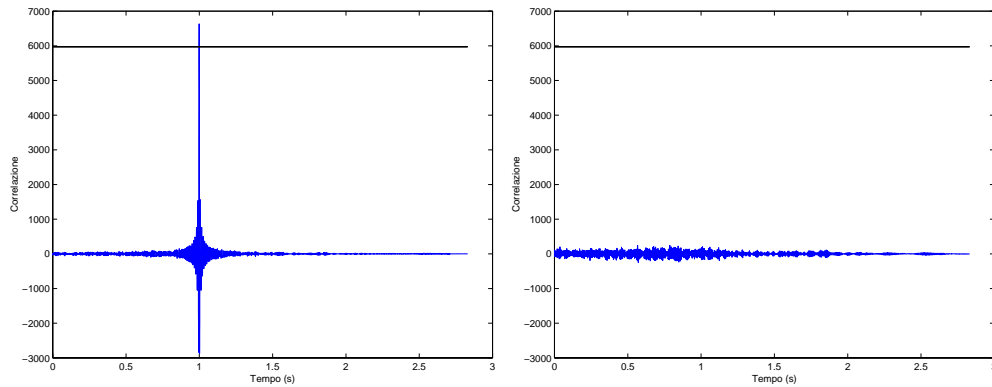


Figura 5.18: Rivelazione corretta sia in presenza che in assenza del segnale

Una volta che è stato identificato l'istante di tempo in cui inizia il segnale, il frame di dati può essere estratto ed analizzato più in dettaglio. Eliminando la parte

di rumore che ha identificato la rete, si può stimare la forma del segnale (figura 5.2.6).

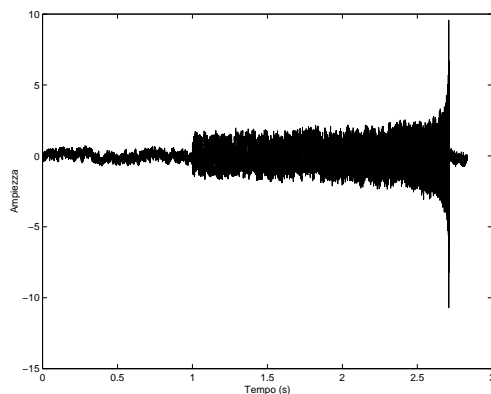


Figura 5.19: Ricostruzione del segnale di ingresso effettuata dalla rete

Sono stati effettuati una serie di esperimenti con diversi segnali (chirp e serie di Lorenz) e pattern di rumore simulato differenti. In figura 5.2.6 possiamo vedere il coefficiente di correlazione fra il segnale estratto ed il segnale originale in funzione del rapporto segnale-rumore. Come si può notare le performances sono molto buone: si mantiene un coefficiente di correlazione intorno a 0.9 anche SNR dell'ordine di -25 , -30 dB

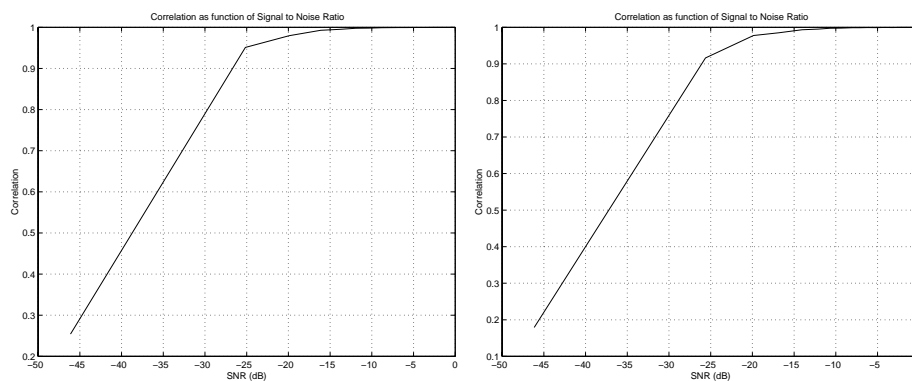


Figura 5.20: Coefficiente di correlazione in funzione del rapporto segnale-rumore per il chirp (sx) e la serie di Lorenz (dx)

5.3 Blind Source Separation per i terremoti esplosivi di Stromboli

La Independent Component Analysis (ICA) è usata per analizzare il segnale sismico prodotto dai terremoti esplosivi del vulcano Stromboli. Abbiamo sperimentalmente provato che è possibile estrarre la componente più significativa dalle registrazioni dei sismometri. In particolare il segnale generato dalla sorgente, che presenta il picco più alto nella densità spettrale, è stato isolato dalle nostre analisi. Nell'ambito dell'esperimento *VIRGO* l'ICA può, come fase di preprocessing, separare i rumori ambientali, per una più corretta identificazione dello spettro di rumore totale.

5.3.1 Origine dei dati

Stromboli è un'isola vulcanica dell'arco delle Eolie, nel mare Tirreno. Il vulcano è alto 924 m sul livello del mare ed il cratere è localizzato 700 m sotto il livello del mare. Questo vulcano è caratterizzato da una attività periodica: avviene una esplosione circa ogni 10 minuti. Le registrazioni presentano un terremoto esplosivo sovrainposto ad un segnale continuo di background, detto tremore. A causa della rapida perdita di coerenza del campo d'onda ed dall'assenza di chiari impulsi, è difficile analizzare questi segnali utilizzando i metodi classici. Le analisi spettrali hanno mostrato la presenza di picchi isolati, connessi alla sorgente.

In particolare è chiara la natura bimodale degli spettri: c'è un rilascio di energie in due bande distinte di frequenza: la prima fra 1 e 3 Hz e l'altra fra i 3 e 5 Hz. La banda 1-3 Hz è connessa all'esistenza di massa oscillante, l'altra banda con la presenza di onde superficiali generate dalla parte superficiale della sorgente e con i meccanismi di scattering distribuiti sull'isola. Lo scopo principale degli studi dei geofisici è isolare queste eccitazioni, cioè la parte del segnale connessa direttamente alla sorgente, come si vede in [1]. I terremoti esplosivi analizzati sono stati registrati

nell'Aprile del 1992 con 15 sismometri elettromagnetici allineati rispetto al cratere in direzione N18S, dall'Università dell'Aquila. La lunghezza tipica di questi segnali, campionati a 125 Hz, è di circa 23 secondi.

5.3.2 Analisi dei dati sismici

La figura 5.21 mostra un esempio di terremoto esplosivo di Stromboli, e la figura 5.22 il suo spettro di potenza.

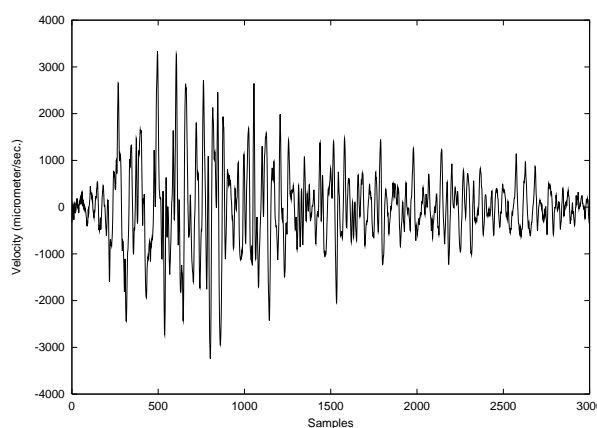


Figura 5.21: Esempio di terremoto esplosivo registrato a Stromboli

Lo spettro rivela una dinamica complessa con molte componenti. I dati consistono nei sismogramma dello stesso terremoto registrato da diverse stazionilocate lungo una linea nei punti LA1-LA15, come si vede in figura 5.23.

Da un precedente lavoro [2], conosciamo la velocità apparente del primo impulso, possiamo quindi stimare i tempi di arrivo del primo impulso ad ogni stazione e sincronizzare le registrazioni. In questo modo le registrazioni risultano misture istantanee dei segnali delle sorgenti, della radiazione di scattering e del rumore ambientali. In base a queste considerazioni possiamo applicare il modello ICA di mistura istantanea.

Innanzitutto abbiamo analizzato gli autovalori della matrice di autocorrelazione

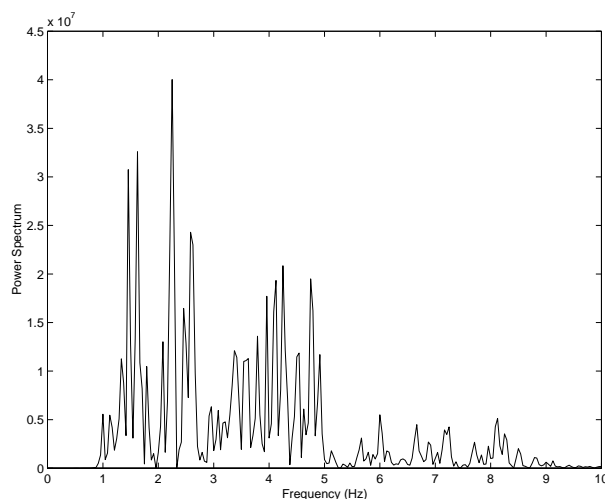


Figura 5.22: Spettro di potenza del segnale graficato in figura 5.21

(vedi figura 5.24), per determinare una stima della informazione contenuta in ogni segnale.

Uno degli scopi dei nostri esperimenti era determinare se ICA deve essere applicata prima o dopo un fase di denoising del segnale. Abbiamo allora separato gli esperimenti in due fasi:

1. abbiamo applicano una tecnica di denoising non lineare e poi è stata usata ICA
2. abbiamo prima separato i segnali indipendenti e poi abbiamo effettuato il denoising

Abbiamo utilizzato una tecnica standard non lineare per eliminare il rumore e la presenza di sistemi dinamici con un attrattore con dimensione maggiore di quella che consideriamo abbia il modello che descrive il terremoto esplosivo. Preferiamo usare questa tecnica anziché i filtri lineari perchè il segnale, generato da sorgenti non lineari, manifesta uno spettro a larga banda e non c'è alcuna giustificazione per considerare una parte dello spettro come segnale. Il denoising non lineare da noi

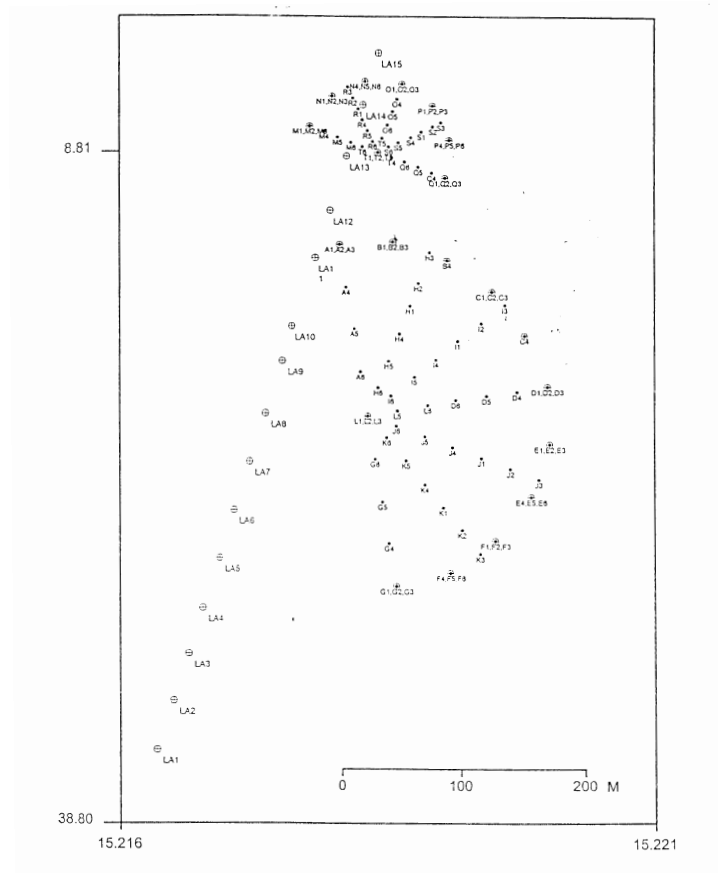


Figura 5.23: Posizione dei sismometri

adottato suppone che il segnale deterministico giace su un sottospazio a bassa dimensione ed il rumore a dimensioni maggiori: allora il denoising consiste nell'identificare questa dimensione e proiettare il segnale nel sottospazio corrispondente [4].

Il denoising non lineare richiede la conoscenza del *time lag*, della dimensione di *embedding* e della dimensione del sottospazio del segnale. I parametri dello spazio delle fasi sono stati calcolati usando le tecniche analizzate nel capitolo II [5]. Abbiamo ottenuto 0.08s per il *time lag* e sei per la dimensione di embedding. In un lavoro precedente [1] è stato provato che la sorgente dei terremoti esplosivi di Stromboli possono essere modellati con un oscillatore autosostenuto il cui attrattore può

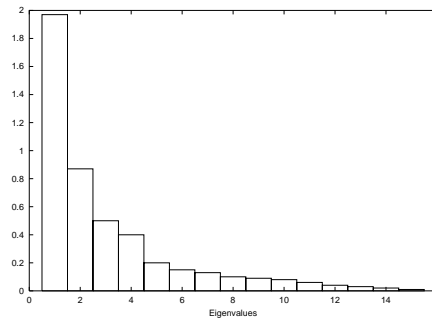


Figura 5.24: Autovalori della matrice di autocorrelazione

essere identificato in uno spazio a dimensione due. Abbiamo allora proiettato le componenti ottenute con la ICA in uno spazio bidimensionale.

5.3.3 Risultati

Sulla base del preprocessing abbiamo estratto sei componenti indipendenti dai segnali dei terremoti esplosivi. Abbiamo effettuato diversi esperimenti concludendo che la fase di denoising elimina parte rilevante del segnale, necessaria affinché si ottenga una buona separazione, come si può notare negli spettri di potenza nella figura 5.25. Concludiamo che nella fase di denoising la perdita di informazione è rilevante per modellare il sistema dinamico che genera il terremoto esplosivo.

Possiamo poi considerare l'ICA come un filtro nel dominio del tempo. Appliciamo prima l'ICA e successivamente la fase di denoising: le sei componenti mantengono tutte le informazioni interessanti, perchè l'algoritmo effettua una separazione ottimale (figura 5.26. Infatti, quattro delle componenti indipendenti (segnali 1,2,3,4 e 6) mostrano un contenuto spettrale differente. Se paragoniamo il contenuto spettrale delle suddette quattro componenti con lo spettro originale di figura 5.22, riconosciamo quattro picchi significativi. La sesta componente, che rileva un picco marcato intorno ad 1 Hz, contiene il 40% della informazione contenuta nelle registrazioni. Questa componente è il segnale diretto della sorgente, come confermano i

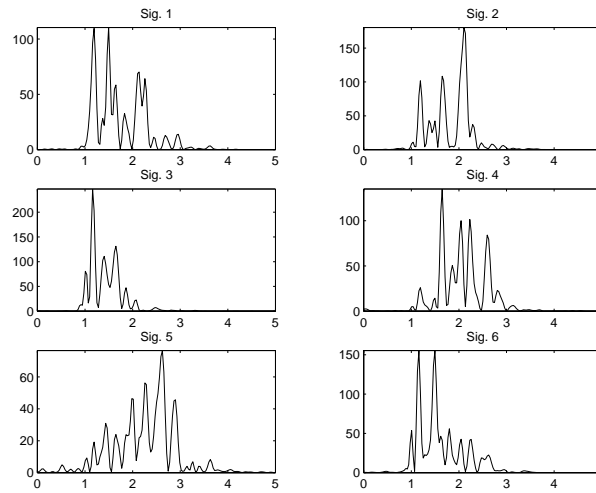


Figura 5.25: Spettro di potenza di ogni componente se i dati sono filtrati prima dell'ICA

risultati in [1].

Abbiamo mostrato che l'ICA è una buona tecnica per analizzare i terremoti esplosivi di Stromboli. Traiamo le seguenti conclusioni:

1. l'ICA deve essere applicata prima di qualsiasi tecnica di denoising che può eliminare informazione utile contenuta nel segnale.
2. Abbiamo estratto segnali significativi utilizzando sei componenti. In particolare, l'applicazione di una tecnica di denoising non lineare alla sesta componente ha chiaramente mostrato un segnale deterministico associato alla sorgente.

Gli esperimenti sono stati effettuati per le tre componenti registrate dai sismometri (radiale, trasversale e verticale) mostrando le stesse caratteristiche. Lo sviluppo futuro prevedere di applicare le stesse metodologie ad altre sorgenti sismiche, come i terremoti tettonici, in cui le registrazioni sono sovrapposizioni di onde che si propagano con diverse velocità. Nell'ambito dell'esperimento *VIRGO* l'ICA

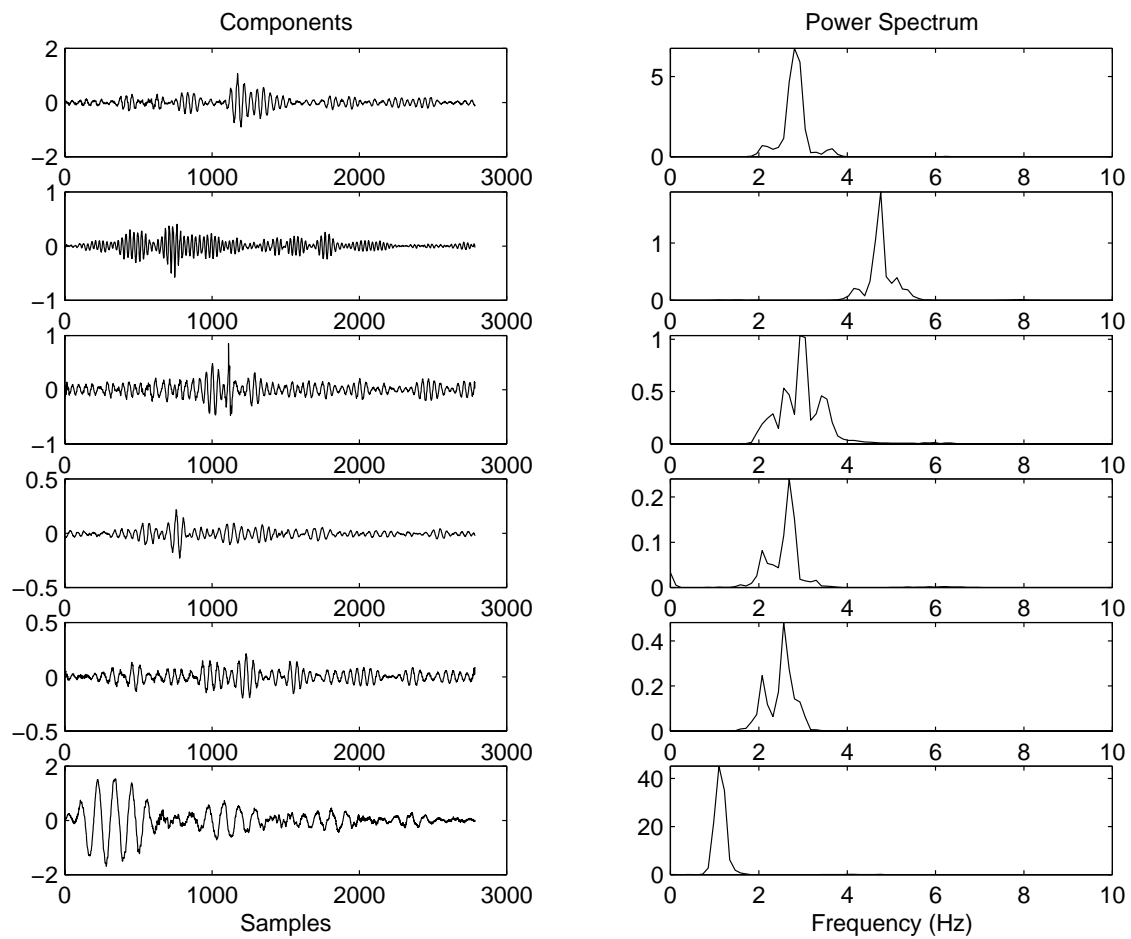


Figura 5.26: Componenti Indipendenti e loro spettri di potenza

può, come fase di preprocessing, separare i rumori ambientali, per una più corretta identificazione dello spettro di rumore totale.

Bibliografia relativa al capitolo V

- [1] S. De Martino, C. Godano, M. Falanga, "Self-sustained oscillator and explosion quakes at Stromboli Volcano" submitted to *Geophys. Res. Lett.*
- [2] B.A. Chouet, G. Saccorotti, M. Martini, P. Dawson, G. De Luca, G. Milana, R. Scarpa, "Source and path effect in wavefield of tremor and explosions at Stromboli Volcano, Italy", *J. Geophys. Res.*, 102, 15129-15150, 1997.
- [3] E. Del Pezzo, C. Godano, A. Gorini, M. Martini, "Wave polarization and location of the source of the explosion quakes at Stromboli Volcano", *IAVCEI Proceedings in Volcanology*, 3, 279-295, Eds. P. Gasparini, R. Scarpa, K. Aki Springer-Verlag Berlin, 1992.
- [4] P. Grassberger, R. Hegger, H. Kanz, C. Schaffrath, T. Schreiber, "On noise reduction methods for chaotic data", *CHAOS*, 3, 127, 1993
- [5] H.D.I. Abarbanel, "Analysis of Observed Chaotic Data", Springer-Verlag, New York, Berlin, 1995
- [6] A.M. Fraser, H.L. Swinney, "Independent coordinates for strange attractors from mutual information", *Phys. Rev. A*, 33, 1134-1139, 1986
- [7] M.B. Kennel, R. Brown, H. D. I. Abarbanel, "Determining embedding dimension for phase space reconstruction using a geometrical construction", *Phys. Rev. A*, 45, 3403-3411, 1992
- [8] A.J. Bell, T.J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution", *Neural Computation*, 7:1129-1159, 1995.

- [9] A. Hyvärinen, E. Oja, "Simple neuron models for independent component analysis", *Proc. 1997 IEEE Int. Conf. on Acoustic, Speech, and Signal Processing (ICASSP'97)*, April 1997, pp. 1483-1492.
- [10] S. Amari, A. Cichocki, H. Yang, "A new learning algorithm for blind signal separation", *Advances in Neural Information Processing System 8*, MIT Press, Cambridge 1996, pp. 757-763.
- [11] J. Karhunen, "Neural approach to independent component analysis and sources separation", *Proc. 4th European Symp. on Artificial Neural Networks (ESANN'96)*, April 1996, pp. 249-266.
- [12] Innocent, "Ultimate sensitivity of Virgo", *Virgonote*, 7-7-94
- [13] Pisani, "Gravitational wave search: real time data analysis strategies on parallel computers", *Virgonote*, 15-3-96
- [14] X. Giannakopoulos, J. Karhunen, E. Oja, "An experimental comparison of neural algorithms for independent component analysis and blind separation", *Int. Jour. on Neural Systems 9(2)*, April 1999, pp. 99-122.
- [15] A. Hyvärinen, "Fast and Robust Fixed-Point Algorithm for Independent Component Analysis", *IEEE Trans. on Neural Networks*, April 1999.
- [16] F. Barone, R. De Rosa, A. Eleuteri, F. Garufi, L. Milano, R. Tagliaferri, "An Neural Networks Based ARX Model of Virgo Noise", in *Proceedings of 11th Workshop on Neural Networks*, Vietri (SA), 99

Conclusioni

Scopo del presente lavoro di tesi è stato sviluppare un metodo rigoroso e versatile per la rivelazione e l'estrazione di segnali gravitazionali nell'ambito dell'esperimento *VIRGO*.

I problemi principali per la rivelazione di segnali gravitazionali sono il basso rapporto segnale rumore e la notevole mole di dati da analizzare. Per tale scopo sono state utilizzate le seguenti tecniche:

- la teoria dei sistemi dinamici non lineari per stimare dimensioni caratteristiche del processo che genera i dati
- la Independent Component Analysis per separare i segnali delle sorgenti dalle registrazioni
- i modelli di elaborazione basati su reti neurali per l'identificazione dei processi
- i matched filter per rivelare la presenza del segnale

In una prima fase abbiamo effettuato una modellizzazione non lineare di alcune variabili ambientali (temperatura e rumore acustico) con una rete neurale. In fase di preprocess abbiamo dovuto stimare il numero ottimale dei ritardi. In base a recenti sviluppi sull'analisi dei sistemi caotici [Abarbanel,1995], abbiamo utilizzato le tecniche di embedding per ricostruire lo spazio delle fasi in cui giaceva la dinamica del sistema che volevamo identificare: il numero di coordinate ritardate dello spazio

di embedding, valutate con il criterio del False Nearest Neighbors, coincide con il numero ottimo di ritardi da utilizzare nell'equazione NARMA che descrive il processo. Abbiamo scelto due reti neurali per effettuare la predizione: una rete di Elman addestrata con le tecniche standard ed una rete con codifica spaziale sull'input con addestramento Bayesiano. I risultati hanno mostrato una superiorità dell'inferenza Bayesiana per l'addestramento delle reti neurali confermata dalla bassa varianza dei residui di predizione rispetto alla varianza dei segnali. Dalla statistica sui residui possiamo determinare le soglie per la rivelazione di anomalie nella variazione delle variabili ambientali.

In una seconda fase ci siamo occupati della modellizzazione non lineare e non parametrica dello spettro di rumore di *VIRGO* nel dominio del tempo. Lo scopo principale era implementare un filtro di whitening per migliorare il matched filter. Da una serie di esperimenti abbiamo concluso che se la durata temporale del data set di addestramento supera i tempi di correlazione fra ingresso ed uscita dell'interferometro, la dinamica è ben identificata anche a basse frequenze. Da precedenti lavori è emerso che una singola rete non riesce a modellare tutte le componenti dello spettro. La tecnica di dividere lo spettro nelle singole sorgenti, modellarle ogniuna con una rete neurale e poi combinare i risultati con i minimi quadrati, ci ha permesso di identificare correttamente anche le risonanze dovute ai modi di violino.

In una terza fase ci siamo occupati della rivelazione e ricostruzione di un segnale gravitazionale simulato, in particolare un segnale chirp generato dalla coalescenza di una sistema binario. Il calcolo del rapporto segnale rumore massimo ci ha permesso di determinare una soglia di rivelazione. L'uso di una rete neurale per implementare il filtro di whitening a monte del matched filter ci ha permesso la corretta rivelazione del segnale e del suo tempo di arrivo all'interferometro. L'identificazione del rumore dell'interferometro nel dominio del tempo con una sola realizzazione del processo stocastico permette inoltre la ricostruzione, nei residui di predizione, della forma

d'onda del segnale.

Nell'ultima fase abbiamo utilizzato la "Independent Component Analysis" per ricostruire i segnali delle sorgenti. Non avendo ancora a disposizione i dati dell'interferometro, abbiamo testato la tecnica sui terremoti esplosivi del vulcano di Stromboli. La registrazione dei sismometri contengono informazioni provenienti dalla sorgente mischiate con segnali riflessi e rifratti della sorgente stessa più altre fonti di rumore. L'applicazione della ICA ci ha permesso di estrarre un segnale deterministico regolare che lavori precedenti [De Martino ed altri] stimano essere il segnale proveniente direttamente dalla sorgente.

Concludendo i risultati raggiunti sono stati i seguenti:

- le tecniche sviluppate nell'analisi dei sistemi caotici permettono di identificare correttamente l'ordine del modello
- l'inferenza Bayesiana in fase di apprendimento è ottimale per il controllo della complessità del modello e per la determinazione delle barre di errore
- la statistica sui residui di predizione permette di rivelare anomalie nelle variazioni delle variabili ambientali. In particolare per la temperatura siamo riusciti a ridurre la varianza dei residui a 0.2 gradi Celsius.
- la struttura con quattro reti neurali identifica correttamente lo spettro di rumore ed è possibile implementare il filtro di whitening
- l'applicazione del matched filter dopo l'identificazione operata dalla rete ci permette di rivelare correttamente l'istante di arrivo del segnale
- abbiamo ricostruito dai residui di predizione del processo stocastico del rumore dell'interferometro la corretta forma del segnale
- l'applicazione dell'ICA ai terremoti di Stromboli ci ha permesso di ricostruire il segnale proveniente direttamente dalla sorgente

Sviluppi futuri per il presente lavoro di tesi sono l'utilizzo dei modelli NAR per il monitoraggio di altre variabili ambientali; l'applicazione dell'ICA ai rumori registrati nel sito dell'interferometro per separare le sorgenti in fase di preprocess; l'allargamento della banda di frequenza identificata dalle reti neurali fino a coprire tutto l'intervallo interessante di osservazione; implementazione di un classificatore basato su reti neurali in sostituzione del matched filter per ottenere la matrice di confusione ed introdurre una funzione rischio nella rivelazione del segnale.

Appendice A

Back Propagation

La funzione di errore *RMS* si può pensare composta dalla somma degli errori ricavati per ciascun pattern di ingresso x^q , per cui si può semplificare la derivazione dell'algoritmo di backpropagation, considerando di calcolare la variazione rispetto ad un singolo pattern di input e alla fine sommando su tutto l'insieme di apprendimento.

$$E = \frac{1}{2} \sum_{q=1}^n \sum_{k=1}^c \{y_k(x^q; \mathbf{w}) - t_k^q\}^2 \quad (\text{A.1})$$

in cui l'indice k indica il neurone di uscita e t_k^q il corrispondente valore desiderato.

Backpropagation cerca il minimo calcolando le derivate del primo ordine rispetto all'errore, per cui, considerando di applicarlo alla rete MLP, si ha per lo strato finale

$$\frac{\partial E^q}{\partial \tilde{w}_{kj}} = \frac{\partial E^q}{\partial \tilde{a}_k} \frac{\partial \tilde{a}_k}{\partial \tilde{w}_{kj}} \quad (\text{A.2})$$

dove con E^q si è indicato l'errore del pattern q .

Se si introduce la definizione di

$$\tilde{\delta}_k = \frac{\partial E^q}{\partial \tilde{a}_k} \quad (\text{A.3})$$

facendo uso della equazione 3.6 si ha che

$$\frac{\partial E^q}{\partial \tilde{w}_{kj}} = \tilde{\delta}_k z_j = [\dot{\tilde{g}}(\tilde{a}_k) \cdot (y(\tilde{a}_k) - t_k)] z_j \quad (\text{A.4})$$

Poichè $\tilde{\delta}_k$ è proporzionale alla differenza tra l'uscita della rete e il valore desiderato viene spesso chiamato errore. Ovviamente il punto sulla funzione di attivazione indica la derivata rispetto all'input netto.

Per quanto riguarda il secondo strato di pesi, possiamo scrivere

$$\frac{\partial E^q}{\partial \tilde{w}_{kj}} = \frac{\partial E^q}{\partial a_k} \frac{\partial a_k}{\partial w_{kj}} \quad (\text{A.5})$$

e ricordando l'equazione 3.6 si ha

$$\frac{\partial E^q}{\partial \tilde{w}_{kj}} = \delta_j x_i \quad (\text{A.6})$$

In questo caso δ_j non è derivabile direttamente in quanto è indirettamente connesso all'errore commesso dalla rete per cui bisogna usare la regola di derivazione di una funzione di funzione, cioè

$$\delta_j = \frac{\partial E^q}{\partial a_j} = \sum_{k=1}^c \frac{\partial E^q}{\partial \tilde{a}_k} \frac{\partial \tilde{a}_k}{\partial a_j} \quad (\text{A.7})$$

e da cui facendo uso delle equazioni precedenti si ha

$$\delta_j = \dot{g}(a_j) \sum_{k=1}^c \tilde{w}_{kj} \tilde{\delta}_k \quad (\text{A.8})$$

Tale espressione può essere interpretata considerando la figura A.1, dove il segnale di errore dei $\tilde{\delta}_k$ è retropropagato lungo i pesi della rete del secondo strato, ed è questa l'origine del termine backpropagation data all'algoritmo.

Infine l'aggiornamento dei pesi è effettuato secondo la regola della discesa secondo gradiente cioè

$$w_{ij}(t) = w_{ij}(t-1) - \eta \frac{\partial E}{\partial w_{ij}} \quad (\text{A.9})$$

in cui w_{ij} è un generico peso di un generico strato, mentre η è detta velocità di apprendimento e regola il passo in cui si procede nell'aggiornamento dei pesi.

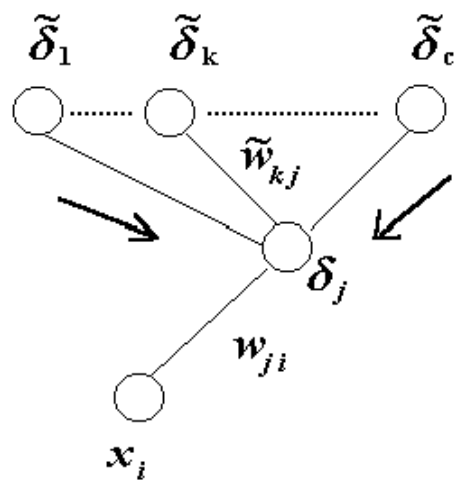


Figura A.1: Diagramma di propagazione dell'errore

Appendice B

Reti con connessioni dinamiche

L'idea di base di queste reti è sostituire le connessioni statiche con filtri dinamici.

Vi sono tre modelli che vengono principalmente utilizzati:

- FIR MLP: consiste del sostituire ogni connessione con un filtro FIR del tipo

$$y(t) = \sum_{i=1}^{n_b} b_i u(t-1) \quad (\text{B.1})$$

dove, per semplicità si può porre n_b uguale per tutte le connessioni e b_i diversi per ogni sinapsi.

- Gamma Network: consiste nel sostituire ogni peso fra l'input e lo strato hidden con un filtro Gamma cosicché

$$y(t) = \left(\frac{\gamma q^{-1}}{1 - \gamma q^{-1}} \right)^i u(t) \quad (\text{B.2})$$

dove $q^{-1}y(t) \triangleq y(t-1)$ è l'operatore di ritardo. γ è costante e i è l'ordine del filtro. Possiamo per convenienza avere filtri dello stesso ordine ma con differente γ per ogni connessione. Se sostituiamo invece le connessioni tra lo strato hidden e quello di output abbiamo l'architettura *Gamma MLP*. Da notare che lo spirito di questo modello è simile alle reti TDNN, tranne che i ritardi sono incorporati tramite un filtro.

- IIR MLP: sono il modello più generale, che incorpora come casi speciali sia le FIR MLP che le Gamma network. Ogni connessione è sostituita da una sinapsi dinamica del tipo filtro IIR:

$$y(t) = \left(\frac{\sum_{i=1}^{n_b} b_i q^{-i}}{1 + \sum_{i=1}^{n_b} a_i q^{-i}} \right) u(t) \quad (\text{B.3})$$

dove $a_i, i = 1, 2, \dots, n_a$ e $b_i, i = 1, 2, \dots, n_b$ sono costanti.

Appendice C

Definizioni di cumulativi di una distribuzione

In questa appendice presenteremo la definizione di cumulativi di ordine superiore. Consideriamo una variabile scalare random con media nulla x , con funzione caratteristica $\hat{f}(t)$:

$$\hat{f}(t) = E\{\exp(itx)\} \quad (\text{C.1})$$

Espandendo il logaritmo della funzione caratteristica in serie di Taylor, otteniamo:

$$\log \hat{f}(t) = k_1(it) + k_2(it)^2/2 + \dots + k_r(it)^r/r! + \dots \quad (\text{C.2})$$

dove k_r sono qualche costante. Queste costanti sono chiamate cumulativi della distribuzione di x . In particolare i primi tre cumulativi (per variabili con media nulla) hanno una espressione semplice:

$$k_1 = E\{x\} \quad (\text{C.3})$$

$$k_2 = E\{x^2\} \quad (\text{C.4})$$

$$k_3 = E\{x^3\} \quad (\text{C.5})$$

Un interesse particolare riveste il cumulativo di ordine quarto, detto *curtosi*,

espresso da

$$kurt(x) = E\{x^4\} - 3(E\{x^2\})^2 \quad (C.6)$$

La curtosi può essere considerata una misura di non gaussianità di x . Per variabili con funzione di distribuzione gaussiana, la curtosi è nulla; tipicamente è positiva per distribuzioni con code più "dure" (super gaussiane) e negativa per distribuzioni con code più "dolci" (sub gaussiane).

I cumulativi possono essere legati ai momenti centrati della distribuzione. Il momento r -esimo di x è definito come $E\{x^r\}$. I momenti possono essere anche ottenuti con una espansione di Taylor della distribuzione $\hat{f}(t)$. Da notare che i primi tre momenti sono identici ai primi tre cumulativi.

La semplicità matematica dell'approccio all'ICA basato sui cumulativi deriva dalla loro proprietà di linearità. Se x_1 e x_2 sono due variabili aleatorie indipendenti, si ha che $kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2)$ e $kurt(\alpha x_1) = \alpha^4 kurt(x_1)$, dove α è uno scalare.

Cumulativi di variabili aleatorie vettoriali $\mathbf{x} = (x_1, \dots, x_m)^T$ si definiscono in modo equivalente in funzione dei coefficienti t_{i_1}, \dots, t_{i_k} della espansione di Taylor della funzione di distribuzione $\hat{f}(t) = E\{\exp(i \sum t_j x_j)\}$.

Ringraziamenti

Vorrei ringraziare tutti coloro che hanno collaborato direttamente o indirettamente a questo lavoro di tesi.

Il ringraziamento più grande è per il mio relatore, il Professore Roberto Tagli-
aferri per avermi formato nel difficile ma stimolante campo delle reti neurali e per
avermi sempre sostenuto in questa fase di maturazione scientifica.

Vorrei sinceramente ringraziare anche in Prof. Leopoldo Milano non solo per i
consigli determinanti che mi ha concesso nel vasto campo dell'analisi dei dati, ma
soprattutto per avermi fatto conoscere quella figura di fisico determinato, versatile e
brillante che tanto cercavo.

Sono grato alla Professoressa Maria Marinaro per avermi offerto la possibilità di
inoltrarmi nello studio della cibernetica.

Un grazie particolare va al Prof. Salvatore De Martino per avermi dedicato nel
corso degli ultimi anni tutta la sua disponibilit  e simpatia e per aver ampliato in
modo determinante la mia conoscenza nel campo dell'analisi dei sistemi non lineari.

Come non poter essere grato alla mia famiglia ed in special modo a mio padre
per aver svegliato in me quel demone della curiosit  che mi ha permesso di non
sentirmi mai appagato di quello che pu  arricchire la mia persona.

Desidero ringraziare tutto il gruppo di ricerca del laboratorio *VIRGO* di Napoli,
Antonio, Fabrizio, Fabio e Angelo, per i preziosi suggerimenti e per aver creato un
ambiente stimolante che mi ha molto aiutato nei momenti di maggior difficolt .

Infine saluto tutti i miei cari compagni di Università che mi hanno accompagnato con ruoli diversi in questi anni indimenticabili: Mariarosaria, Angelo, Gabriella, Giulia, Samanta, Paola, Nicola, Maria, Palma, Carlo, Eugenio, Paola Cusano, Annalisa, Mimmo, Arturo, e tanti altri, con cui ho condiviso negli ultimi anni tutti i momenti più importanti della mia crescita, sia culturale che umana. Dedico un abbraccio affettuoso a Lara, con cui ho trascorso quasi tutte le mattine ed i pomeriggi quando viaggiavo per raggiungere l'università. Quando tutto intorno a me cambiava lei era sempre lì, alle 8:30, che mi accompagnava verso un'altra giornata di studio.

Un sorriso di gratitudine ad Henry Haller, ch  ha impresso in modo determinante la direzione della mia vita.

Fausto Acernese